

**IMPROVED COMPUTING ARCHITECTURE AND RELATED SYSTEM AND METHOD**

Publication number: JP2006518058 (T)

Publication date: 2006-06-03

Inventor(s):

Applicant(s):

Classification:






- International: G06F15/80; G06F9/30; G06F9/38; G06F9/445; G06F9/46; G06F15/78; G06F15/76; G06F9/30; G06F9/38; G06F9/445; G06F9/46

- European: G06F9/3854

Application number: JP20050502225T 20031031

Priority number(s): US20030422503P 20021031; US20030683929 20031009; US20030683932 20031009; US20030684053 20031009; US20030684057 20031009; US20030684102 20031009; WO20030634558 20031031

Also published as:

 WO2004042560 (A2) WO2004042560 (A3) WO2004042574 (A2) WO2004042574 (A3) WO2004042569 (A2)

more &gt;&gt;

Abstract not available for JP 2006518058 (T)

Abstract of corresponding document: **WO 2004042560 (A2)**

A peer-vector machine includes a host processor and a hardwired pipeline accelerator. The host processor executes a program, and, in response to the program, generates host data, and the pipeline accelerator generates pipeline data from the host data. Alternatively, the pipeline accelerator generates the pipeline data, and the host processor generates the host data from the pipeline data. Because the peer-vector machine includes both a processor and a pipeline accelerator, it can often process data more efficiently than a machine that includes only processors or only accelerators. For example, one can design the peer-vector machine so that the host processor performs decision-making and non-mathematically intensive operations and the accelerator performs non-decision-making and mathematically intensive operations. By shifting the mathematically intensive operations to the accelerator, the peer-vector machine often can, for a given clock frequency, process data at a speed that surpasses the speed at which a processor-only machine can process the data.

Data supplied from the **esp@cenet** database — Worldwide

(19) 日本国特許庁(JP)

(12) 公表特許公報(A)

(11) 特許出願公表番号

特表2006-518058

(P2006-518058A)

(43) 公表日 平成18年8月3日(2006.8.3)

(51) Int. Cl.

G06F 15/80 (2006.01)

F I

G06F 15/80

テーマコード (参考)

審査請求 未請求 予備審査請求 未請求 (全 44 頁)

(21) 出願番号	特願2005-502225 (P2005-502225)	(71) 出願人	504242618
(86) (22) 出願日	平成15年10月31日 (2003.10.31)		ロッキード マーティン コーポレーション
(85) 翻訳文提出日	平成17年6月16日 (2005.6.16)		アメリカ合衆国 メリーランド州 208
(86) 国際出願番号	PCT/US2003/034558		17-1803 ベセスダ ロックレッジ
(87) 国際公開番号	W02004/042562		ドライブ 6801
(87) 国際公開日	平成16年5月21日 (2004.5.21)	(74) 代理人	100083932
(31) 優先権主張番号	60/422,503		弁理士 廣江 武典
(32) 優先日	平成14年10月31日 (2002.10.31)	(74) 代理人	100129698
(33) 優先権主張国	米国 (US)		弁理士 武川 隆寛
(31) 優先権主張番号	10/683,929	(74) 代理人	100129676
(32) 優先日	平成15年10月9日 (2003.10.9)		弁理士 ▲高▼荒 新一
(33) 優先権主張国	米国 (US)	(74) 代理人	100130074
(31) 優先権主張番号	10/683,932		弁理士 中村 繁元
(32) 優先日	平成15年10月9日 (2003.10.9)		
(33) 優先権主張国	米国 (US)		

最終頁に続く

(54) 【発明の名称】 改善された計算アーキテクチャ用パイプライン加速器、関連システム、並びに、方法

## (57) 【要約】

パイプライン加速器はメモリと該メモリと結合されたハードウェアに組み込まれたパイプライン回路とを含む。このハードウェアに組み込まれたパイプライン回路は、データを受信し、前記データを前記メモリにロードし、前記メモリから前記データを検索し、前記検索されたデータを処理し、そして前記処理されたデータを外部ソースに提供するように動作できる。加えて或は代替例で、ハードウェアに組み込まれたパイプライン回路は、データを受信し、前記受信されたデータを処理し、前記処理されたデータを前記メモリにロードし、前記メモリから前記処理されたデータを検索し、そして前記検索された処理データを外部ソースに提供するように動作できる。パイプライン加速器がピアベクトル・マシンの一部としてのプロセッサと結合されている場合、前記メモリは前記ハードウェアに組み込まれたパイプライン回路と前記プロセッサが実行するアプリケーションとの間におけるデータの転送一方向性であろうが二方向性であろうが一を補助する。

10

## 【特許請求の範囲】

## 【請求項1】

パイプライン加速器であって、  
メモリと、  
前記メモリと結合されたハードウェアに組み込まれたパイプライン回路と、を含み、  
前記パイプライン回路が、  
データを受信し、  
前記データを前記メモリにロードし、  
前記メモリから前記データを検索し、  
前記検索されたデータを処理し、  
前記処理されたデータを外部ソースに提供するように動作できる、パイプライン加速器。

10

## 【請求項2】

前記メモリが第1集積回路上に配置され、  
前記パイプライン回路が第2集積回路上に配置されている、請求項1に記載のパイプライン加速器。

## 【請求項3】

前記パイプライン回路がフィールド・プログラマブル・ゲート・アレイ上に配置されている、請求項1に記載のパイプライン加速器。

## 【請求項4】

前記パイプライン回路が、  
前記処理されたデータを前記メモリにロードし、  
前記メモリから前記処理されたデータを検索し、  
前記検索された処理データを前記外部ソースに提供することによって、前記処理されたデータを前記外部ソースに提供するように動作できる、請求項1に記載のパイプライン加速器。

20

## 【請求項5】

前記外部ソースがプロセッサを含み、  
前記パイプライン回路が前記プロセッサから前記データを受信するように動作できる、  
請求項1に記載のパイプライン加速器。

30

## 【請求項6】

計算マシンであって、  
プロセッサと、  
前記プロセッサと結合されたパイプライン加速器と、を含み、  
前記パイプライン加速器が、  
メモリと、  
前記メモリと結合されたハードウェアに組み込まれたパイプライン回路と、を含み、  
前記パイプライン回路が、  
前記プロセッサからデータを受信し、  
前記データを前記メモリにロードし、  
前記メモリから前記データを検索し、  
前記検索されたデータを処理し、  
前記処理されたデータを前記プロセッサに提供するように動作できる、計算マシン。

40

## 【請求項7】

パイプライン加速器であって、  
メモリと、  
前記メモリと結合されたハードウェアに組み込まれたパイプライン回路と、を含み、  
前記パイプライン回路が、  
データを受信し、

50

前記受信されたデータを処理し、  
前記処理されたデータを前記メモリにロードし、  
前記メモリから前記処理されたデータを検索し、  
前記検索された処理データを外部ソースに提供するように動作できる、パイプライン  
加速器。

【請求項 8】

計算マシンであって、  
プロセッサと、  
前記プロセッサと結合されたパイプライン加速器と、を含み、  
前記パイプライン加速器が、  
メモリと、  
ハードウェアに組み込まれたパイプライン回路と、を含み、  
前記ハードウェアに組み込まれたパイプライン回路が、  
前記プロセッサからデータを受信し、  
前記受信されたデータを処理し、  
前記処理されたデータを前記メモリにロードし、  
前記メモリから前記処理されたデータを検索し、  
前記検索された処理データを前記プロセッサに提供するように動作できる、計  
算マシン。

【請求項 9】

パイプライン加速器であって、  
第 1 及び第 2 のメモリと、  
前記第 1 及び第 2 のメモリと結合されたハードウェアに組み込まれたパイプライン回路  
と、を含み、  
前記パイプライン回路が、  
外部ソースから生データを受信してその生データを前記第 1 メモリにロードするよう  
に動作できる入力データ・ハンドラーと、  
前記生データを処理するように動作できるハードウェアに組み込まれたパイプライン  
と、  
前記第 1 メモリから前記生データを検索し、その検索された生データを前記ハードウ  
ェアに組み込まれたパイプラインに提供してから、前記ハードウェアに組み込まれたパイ  
プラインから前記第 2 メモリに処理されたデータをロードするように動作できるパイプラ  
イン・インターフェースと、  
前記第 2 メモリから前記処理されたデータを検索してその処理されたデータを前記外  
部ソースに提供するように動作できる出力データ・ハンドラーと、を含む、パイプライン  
加速器。

【請求項 10】

前記第 1 及び第 2 のメモリの各々がそれぞれの第 1 及び第 2 のポートを含み、  
前記入力データ・ハンドラーが、前記第 1 メモリの前記第 1 ポートを介して、前記生デ  
ータをロードするように動作でき、  
前記パイプライン・インターフェースが、前記第 1 メモリの前記第 2 ポートを介して、  
前記生データを検索して前記処理されたデータを前記第 2 メモリの前記第 1 ポートを介し  
てロードするように動作でき、  
前記出力データ・ハンドラーが、前記第 2 メモリの前記第 2 ポートを介して、前記処理  
されたデータを検索するように動作できる、請求項 9 に記載のパイプライン加速器。

【請求項 11】

前記ハードウェアに組み込まれたパイプライン回路と結合された第 3 メモリを更に含み、  
前記ハードウェアに組み込まれたパイプラインが、前記生データを処理しながら中間デ  
ータを生成するように動作でき、

前記パイプライン・インターフェースが、前記中間データを前記第3メモリにロードして前記第3メモリから前記中間データを検索するように動作できる、請求項9に記載のパイプライン加速器。

【請求項12】

前記第1及び第2のメモリが第1及び第2の集積回路上にそれぞれ配置され、

前記パイプライン回路がフィールド・プログラマブル・ゲート・アレイ上に配置されている、請求項9に記載のパイプライン加速器。

【請求項13】

前記入力データ・ハンドラー及び前記パイプライン・インターフェースと結合されている入力データ・キューを更に含み、

前記入力データ・ハンドラーが、前記第1メモリ内の前記生データの箇所を指すポイントを前記入力データ・キューにロードし、

前記パイプライン・インターフェースが前記箇所から前記ポイントを用いて前記生データを検索するように動作できる、請求項9に記載のパイプライン加速器。

【請求項14】

前記出力データ・ハンドラー及び前記パイプライン・インターフェースと結合された出力データ・キューを更に含み、

前記パイプライン・インターフェースが、前記第2メモリ内の前記処理データの箇所を指すポイントを前記出力データ・キューにロードし、

前記出力データ・ハンドラーが、前記箇所から前記ポイントを用いて前記処理データを検索するように動作できる、請求項9に記載のパイプライン加速器。

【請求項15】

前記入力データ・ハンドラー、ハードウェアに組み込まれたパイプライン、パイプライン・インターフェース、並びに、出力データ・ハンドラーが各動作コンフィギュレーションを有し、

前記入力データ・ハンドラー、ハードウェアに組み込まれたパイプライン、パイプライン・インターフェース、並びに、出力データ・ハンドラーと結合されると共に、それらの前記動作コンフィギュレーションを設定するように動作できるコンフィギュレーション・マネージャを更に含む、請求項9に記載のパイプライン加速器。

【請求項16】

前記入力データ・ハンドラー、ハードウェアに組み込まれたパイプライン、パイプライン・インターフェース、並びに、出力データ・ハンドラーが各動作状況を有し、

前記入力データ・ハンドラー、ハードウェアに組み込まれたパイプライン、パイプライン・インターフェース、並びに、出力データ・ハンドラーと結合されると共に、動作状況に応じて、それらにおける例外を識別するように動作できる例外マネージャを更に含む、請求項9に記載のパイプライン加速器。

【請求項17】

パイプライン加速器であって、

データを処理するように動作できるハードウェアに組み込まれたパイプラインと、

前記ハードウェアに組み込まれたパイプラインと結合された入力データ・ハンドラーと、

前記入力データ・ハンドラーが、

前記データを受信し、

前記データが前記ハードウェアに組み込まれたパイプラインに向けられているかを決定し、

前記データが前記ハードウェアに組み込まれたパイプラインに向けられていれば、そのデータをそのハードウェアに組み込まれたパイプラインに提供するように動作できる、パイプライン加速器。

【請求項18】

前記入力データ・ハンドラーが、

ヘッダー及び前記データを含むメッセージを受信して、前記メッセージから前記データを抽出することによって前記データを受信し、

前記ヘッダーを分析することによって前記データが前記ハードウェアに組み込まれたパイプラインに向けられているかを決定するように更に動作できる、請求項17に記載のパイプライン加速器。

【請求項19】

前記ハードウェアに組み込まれたパイプライン及び前記入力データ・ハンドラーが単一のフィールド・プログラマブル・ゲート・アレイ上に配置されている、請求項17に記載のパイプライン加速器。

【請求項20】

前記ハードウェアに組み込まれたパイプライン及び前記入力データ・ハンドラーが各フィールド・プログラマブル・ゲート・アレイ上にそれぞれ配置されている、請求項17に記載のパイプライン加速器。

【請求項21】

計算マシンであって、

プロセッサと、

前記プロセッサと結合されたパイプライン加速器と、を含み、

前記パイプライン加速器が、

データを処理するように動作できるハードウェアに組み込まれたパイプラインと、

前記ハードウェアに組み込まれたパイプラインと結合されると共に、前記プロセッサから前記データを受信し、前記データが前記ハードウェアに組み込まれたパイプラインに向けられているかを決定し、もし前記データが前記ハードウェアに組み込まれたパイプラインに向けられていれば、前記データを前記ハードウェアに組み込まれたパイプラインに提供するように動作できる入力データ・ハンドラーと、含む、計算マシン。

【請求項22】

パイプライン加速器であって、

データを生成するように動作できるハードウェアに組み込まれたパイプラインと、

前記ハードウェアに組み込まれたパイプラインと結合された出力データ・ハンドラーと、を含み、

前記出力データ・ハンドラーが、

前記データを受信し、

前記データの仕向先を決定し、

前記データを前記仕向先に提供するように動作できる、パイプライン加速器。

【請求項23】

前記出力データ・ハンドラーが、

前記データのタイプを識別して、前記データの前記タイプに基づき前記仕向先を決定することによって前記データの仕向先を決定するように動作できると共に、

前記仕向先を識別し且つ前記データを含むメッセージを生成して、前記メッセージを前記仕向先に提供するように動作できる、請求項22に記載のパイプライン加速器。

【請求項24】

計算マシンであって、

アプリケーションのスレッドを実行するように動作できるプロセッサと、

前記プロセッサと結合されたパイプライン加速器と、を含み、

前記パイプライン加速器が、

データを生成するように動作できるハードウェアに組み込まれたパイプラインと、

前記ハードウェアに組み込まれたパイプラインと結合された出力データ・ハンドラーと、を含み、

前記出力データ・ハンドラーが、

前記データを受信し、

前記データに加入する前記アプリケーションのスレッドを識別し、

前記データを前記加入スレッドに提供するように動作できる、計算マシン。

【請求項 25】

パイプライン加速器であって、

データ値を処理するように動作できるハードウェアに組み込まれたパイプラインと、

前記ハードウェアに組み込まれたパイプラインと結合されると共に、該ハードウェアに組み込まれたパイプラインの動作を制御するように動作できるシーケンス・マネージャと、を含むパイプライン加速器。

【請求項 26】

前記シーケンス・マネージャが前記ハードウェアに組み込まれたパイプラインが前記データ値を受信する順番を制御するように動作できる、請求項 25 に記載のパイプライン加速器。

10

【請求項 27】

前記シーケンス・マネージャが、

事象を受信し、

前記事象に応じて前記ハードウェアに組み込まれたパイプラインを制御するように更に動作できる、請求項 25 に記載のパイプライン加速器。

【請求項 28】

前記シーケンス・マネージャが、

同期信号を受信し、

前記同期信号に応じて前記ハードウェアに組み込まれたパイプラインの動作を制御するように更に動作できる、請求項 25 に記載のパイプライン加速器。

20

【請求項 29】

前記シーケンス・マネージャが、

前記ハードウェアに組み込まれたパイプラインに対する実現値を検知し、

前記実現値に応じて事象を生成するように更に動作できる、請求項 25 に記載のパイプライン加速器。

【請求項 30】

計算マシンであって、

データ及び事象を生成するように動作できるプロセッサと、

前記プロセッサと結合されたパイプライン加速器と、を含み、

前記パイプライン加速器が、

前記プロセッサから前記データを受信してその受信されたデータを処理するように動作できるハードウェアに組み込まれたパイプラインと、

前記ハードウェアに組み込まれたパイプラインと結合されると共に、前記プロセッサから前記事象を受信して前記事象に応じて前記ハードウェアに組み込まれたパイプラインの動作を制御するように動作できるシーケンス・マネージャと、を含む、計算マシン。

30

【請求項 31】

パイプライン加速器であって、

動作コンフィギュレーションを有すると共にデータを処理するように動作できるハードウェアに組み込まれたパイプライン回路と、

前記ハードウェアに組み込まれたパイプライン回路と結合されると共に、前記動作コンフィギュレーションを設定するように動作できるコンフィギュレーション・マネージャと、を含むパイプライン加速器。

40

【請求項 32】

前記ハードウェアに組み込まれたパイプライン回路がコンフィギュレーション・レジスタを含み、

前記コンフィギュレーション・マネージャが前記コンフィギュレーション・レジスタにコンフィギュレーション値をロードすることによって前記動作コンフィギュレーションを設定するように動作できる、請求項 31 に記載のパイプライン加速器。

【請求項 33】

50

前記コンフィギュレーション・マネージャが外部ソースから前記コンフィギュレーション値を受信するように動作できる、請求項32に記載のパイプライン加速器。

【請求項34】

計算マシンであって、

データ及びコンフィギュレーション値を生成するように動作できるプロセッサと、

前記プロセッサと結合されたパイプライン加速器と、を含み、

前記パイプライン加速器が、

動作コンフィギュレーションを有すると共に前記データを処理するように動作できるハードウェアに組み込まれたパイプライン回路と、

前記ハードウェアに組み込まれたパイプライン回路と結合されると共に、前記コンフィギュレーション値に応じて前記動作コンフィギュレーションを設定するように動作できるコンフィギュレーション・マネージャと、を含む、計算マシン。

10

【請求項35】

パイプライン加速器であって、

動作状況を有すると共にデータを処理するように動作できるハードウェアに組み込まれたパイプライン回路と、

前記ハードウェアに組み込まれた回路と結合されると共に、前記動作状況に応じて前記ハードウェアに組み込まれた回路の前記動作状況における例外を識別するように動作できる例外マネージャと、

を含むパイプライン加速器。

20

【請求項36】

前記ハードウェアに組み込まれたパイプライン回路が前記動作状況を表す状況値を生成するように動作でき、

前記例外マネージャが前記状況値に応じて前記例外を識別するように動作できる、請求項35に記載のパイプライン加速器。

【請求項37】

前記ハードウェアに組み込まれたパイプライン回路が前記状況値を記憶するように動作できる状況レジスタを含み、

前記例外マネージャが前記状況レジスタから前記状況値を受信する、請求項36に記載のパイプライン加速器。

30

【請求項38】

前記例外マネージャが、外部ソースへの前記ハードウェアに組み込まれたパイプライン回路の前記動作状況における例外を識別するように動作できる、請求項35に記載のパイプライン加速器。

【請求項39】

計算マシンであって、

データを生成するように動作できるプロセッサと、

パイプライン加速器と、を含み、

前記パイプライン加速器が、

動作状況を有すると共にデータを処理して前記動作状況を表す状況値を生成するように動作できるハードウェアに組み込まれたパイプライン回路と、

40

前記ハードウェアに組み込まれたパイプライン回路と結合されると共に、前記状況値に応じて前記ハードウェアに組み込まれたパイプライン回路の前記動作状況における例外を識別して前記プロセッサに前記例外を通知するように動作できる例外マネージャと、を含む、計算マシン。

【請求項40】

計算マシンであって、

動作状況を有してデータを処理するように動作できるハードウェアに組み込まれたパイプライン回路と、前記ハードウェアに組み込まれたパイプライン回路と結合されると共に、前記動作状況を表す状況値を生成するように動作できる例外マネージャと、を含むパイ

50



ブライン加速器と、

前記パイプライン加速器と結合されると共に、前記データを生成し、前記状況値を受信してから、前記状況値を分析することによって前記ハードウェアに組み込まれたパイプライン回路が動作しているかを決定するように動作できるプロセッサと、を含む、計算マシン。

【請求項 4 1】

方法であって、

データをメモリにロードし、

前記メモリから前記データを検索し、

ハードウェアに組み込まれたパイプライン回路によって前記検索されたデータを処理し 10

、前記処理されたデータを外部ソースに提供することを含む方法。

【請求項 4 2】

前記処理されたデータを提供することが、

前記処理されたデータを前記メモリにロードし、

前記メモリから前記処理されたデータを検索し、

前記検索された処理データを前記外部ソースに提供することを含む、請求項 4 1 に記載の方法。

【請求項 4 3】

方法であって、

ハードウェアに組み込まれたパイプライン回路によってデータを処理し、

前記処理されたデータをメモリにロードし、

前記メモリから前記処理されたデータを検索し、

前記検索された処理データを外部ソースに提供することを含む、方法。

20

【請求項 4 4】

方法であって、

外部ソースから第 1 メモリに生データをロードし、

前記第 1 メモリから前記生データを検索し、

ハードウェアに組み込まれたパイプラインによって前記検索されたデータを処理し、

前記ハードウェアに組み込まれたパイプラインから第 2 メモリに前記処理されたデータ 30

をロードし、前記第 2 メモリから前記外部ソースに前記処理されたデータを提供することを含む、方法。

【請求項 4 5】

前記生データをロードすることが、前記第 1 メモリの第 1 ポートを介して前記生データをロードすることを含み、

前記生データを検索することが、前記第 1 メモリの第 2 ポートを介して前記生データを検索することを含み、

前記処理されたデータをロードすることが、前記第 2 メモリの第 1 ポートを介して前記処理されたデータをロードすることを含み、

40

前記処理されたデータを提供することが、前記第 2 メモリの第 2 ポートを介して前記処理されたデータを検索することを含む、請求項 4 4 に記載の方法。

【請求項 4 6】

前記生データの処理に応じて前記ハードウェアに組み込まれたパイプラインによって中間データを生成し、

前記中間データを第 3 メモリにロードし、

前記第 3 メモリから前記ハードウェアに組み込まれたパイプラインに戻すように前記中間データを提供することを含む、請求項 4 4 に記載の方法。

【請求項 4 7】

前記第 1 メモリ内の前記生データの箇所を指すポインタを人力メッセージ・キューにロ 50

ードすることを更に含み、

前記生データを検索することが、前記箇所から前記ポインタを用いて前記生データを検索することを含む、請求項44に記載の方法。

【請求項48】

前記第2メモリ内の前記処理されたデータの箇所を指すポインタを出力メッセージ・キューにロードすることを更に含み、

前記処理されたデータを検索することが、前記箇所から前記ポインタを用いて前記処理されたデータを検索することを含む、請求項44に記載の方法。

【請求項49】

前記生データをロードして検索し、その検索されたデータを処理して、その処理されたデータをロードして提供するためのパラメータを設定することを更に含み、請求項44に記載の方法。 10

【請求項50】

前記生データの前記ロード及び前記検索、前記検索されたデータの前記処理、並びに、前記処理されたデータの前記ロード及び前記提供の間にエラーが生ずるかを決定することを更に含み、請求項44に記載の方法。

【請求項51】

方法であって、

データを受信し、

前記データがハードウェアに組み込まれたパイプラインに向けられているかを決定し、 20  
もし前記データが前記ハードウェアに組み込まれたパイプラインに向けられていれば前記ハードウェアに組み込まれたパイプラインに前記データを提供することを含む方法。

【請求項52】

前記データを受信することが、

ヘッダー及び前記データを含むメッセージを受信し、

前記メッセージから前記データを抽出することを含み、

前記データが前記ハードウェアに組み込まれたパイプラインに向けられているかを決定することが、前記ヘッダーを分析することを含む、請求項51に記載の方法。

【請求項53】

方法であって、

ハードウェアに組み込まれたパイプラインによってデータを生成し、

前記データの仕向先を決定し、

前記データを前記仕向先に提供することを含む方法。 30

【請求項54】

前記データの前記仕向先を決定することが、

前記データのタイプを識別し、

前記データの前記タイプに基づき前記仕向先を決定することを含み、

前記データを前記仕向先に提供することが、

前記仕向先を識別すると共に前記データを含むメッセージを生成し、

前記メッセージを前記仕向先に提供することを含む、請求項53に記載の方法。 40

【請求項55】

方法であって、

ハードウェアに組み込まれたパイプラインによってデータ値を処理し、

前記ハードウェアに組み込まれたパイプラインの動作を順序付けすることを含む方法。

【請求項56】

前記動作を順序付けることが、

前記ハードウェアに組み込まれたパイプラインが前記データ値を処理する順番を順序付けることを含む、請求項55に記載の方法。

【請求項57】

前記動作を順序付けることが、 50

前記ハードウェアに組み込まれたパイプラインを同期信号と同期させることを含む、請求項55に記載の方法。

【請求項58】

前記ハードウェアに組み込まれたパイプラインの動作中に予め規定された実現値を検知し、

前記実現値に応じて事象を生成することを更に含む、請求項55に記載の方法。

【請求項59】

方法であって、

コンフィギュレーション値をレジスタにロードし、

前記コンフィギュレーション値によってハードウェアに組み込まれたパイプラインの動作コンフィギュレーションを設定することを含む方法。 10

【請求項60】

方法であって、

ハードウェアに組み込まれたパイプラインによってデータを処理し、

前記ハードウェアに組み込まれたパイプラインの動作状況进行分析することによって前記処理されたデータにおけるエラーを識別することを含む方法。

【請求項61】

ハードウェアに組み込まれたパイプライン回路を設計する方法であって、

ライブラリから通信インターフェースの第1データ表現を検索し、

前記通信インターフェースと結合されることになるハードウェアに組み込まれたパイプラインの第2データ表現を生成し、 20

前記第1及び第2のデータ表現を組み合わせて、前記ハードウェアに組み込まれたパイプライン回路に対するハード・コンフィギュレーション・データを生成することを含む方法。

【請求項62】

前記第1及び第2のデータ表現を組み合わせる前に、前記サービス層の所定パラメータに対する値を選択することによって前記第1データ表現を変更することを更に含む、請求項61に記載の方法。

【請求項63】

前記通信インターフェースが、前記ハードウェアに組み込まれたパイプライン回路に別の回路と通信させるように動作できる、請求項61に記載の方法。 30

【請求項64】

前記第1及び第2のデータ表現を組み合わせることが、前記第1及び第2のデータ表現を前記ハード・コンフィギュレーション・データに組み合わせるを含む、請求項61に記載の方法。

【請求項65】

前記ハード・コンフィギュレーション・データがファームウェアを含む、請求項61に記載の方法。

【発明の詳細な説明】

【技術分野】

40

【0001】

<優先権の請求>

この出願は、下記の特許文献1に対する優先権を請求するものであり、引用することでここに合体させる。

【特許文献1】米国仮出願第60/422,503号(2002年10月31日出願)

【0002】

<関連出願の相互参照>

この出願は、「改善された計算アーキテクチャ、関連システム、並びに、方法」と題された下記の特許文献2、「改善された計算アーキテクチャを有する計算マシン、関連システム、並びに、方法」と題された下記の特許文献3、「プログラマブル回路、関連計算マ 50

シン、並びに、方法」と題された下記の特許文献4、「多数パイプライン・ユニットを有するパイプライン加速器、関連計算マシン、並びに、方法」と題された下記の特許文献5と関連し、これら特許文献は全て2003年10月9日に出願され、共通の所有者を有し、引用することでここに合体させる。

【特許文献2】米国出願第10/684,102号

【特許文献3】米国出願第10/684,053号

【特許文献4】米国出願第10/684,057号

【特許文献5】米国出願第10/683,932号

【背景技術】

【0003】

比較的大量のデータを比較的短い期間で処理する通常の計算アーキテクチャは、処理負担を分担する多数の相互接続プロセッサを含む。処理負担を分担することによって、これら多数のプロセッサは、しばしば、所与のクロック周波数で単一プロセッサができるものよりよりも迅速にデータを処理できる。例えば、これらプロセッサの各々はデータの各部分を処理できるか、或は、処理アルゴリズムの各部分を実行できる。

【0004】

図1は、多数プロセッサ・アーキテクチャを有する従来の計算マシン10の概略ブロック図である。この計算マシン10は、マスター・プロセッサ12と、相互に通信すると共に該マスター・プロセッサとバス16を介して通信する共同プロセッサ14、…14、と、遠隔装置（図1では不図示）から生データを受け取る入力ポート18と、該遠隔装置に処理データを提供する出力ポート20とを含む。また、計算マシン10はマスター・プロセッサ12に対するメモリ22と、共同プロセッサ14、…14、に対する各メモリ24、…24、と、マスター・プロセッサ及び共同プロセッサがバス16を介して共有するメモリ26とを含む。メモリ22はマスター・プロセッサ12に対するプログラム及び作業メモリの双方の役割を果たし、各メモリ24、…24、は各共同メモリ14、…14、に対するプログラム及び作業メモリの双方の役割を果たす。共有されたメモリ26は、マスター・プロセッサ12及び共同プロセッサ14がそれらの間でデータを転送すること、ポート18を介して遠隔装置からデータを転送すること、ポート20を介して遠隔装置にデータを転送することを可能としている。またマスター・プロセッサ12及び共同プロセッサ14は、マシン10が生データを処理する速度を制御する共通クロック信号を受け取る。

【0005】

一般に、計算マシン10は、マスター・プロセッサ12及び共同プロセッサ14の間で生データの処理を効果的に分割する。ソナー・アレイ等の遠隔ソース（図1では不図示）は、ポート18を介して、生データに対する先入れ先出し（FIFO）バッファ（不図示）として作用する共有メモリ26の1つの区分に生データをロードする。マスター・プロセッサ12はバス16を介してメモリ26から生データを検索して、マスター・プロセッサ及び共同プロセッサ14はその生データを処理して、バス16を介して必要に応じてデータをそれらの間に転送する。マスター・プロセッサ12はその処理データを共有メモリ26内に規定された別のFIFOバッファ（不図示）にロードし、遠隔ソースがポート20を介してこのFIFOからその処理データを検索する。

【0006】

演算例において、計算マシン10は生データに対する $n+1$ 個の各演算を順次実行することによって該生データを処理し、これら演算は一体的に高速フーリエ変換（FFT）等の処理アルゴリズムを構成する。より詳細には、マシン10はマスター・プロセッサ12及び共同プロセッサ14からのデータ-処理パイプラインを形成する。クロック信号の所与の周波数で、そうしたパイプラインはしばしばマシン10が単一プロセッサのみを有するマシンよりも高速に生データを処理することを可能としている。

【0007】

メモリ26内における生データFIFO（不図示）からの生データ検索後、マスター・プロセッサ12はその生データに対して三角関数等の第1番演算を実行する。この演算は

第1番結果を生み出し、それをプロセッサ12がメモリ26内に規定された第1番結果F1FO（不図示）に記憶する。典型的には、プロセッサ12はメモリ22内に記憶されたプログラムを実行し、そのプログラムの制御の下で上述した動作を実行する。プロセッサ12はメモリ22を作業メモリとしても使用し得て、当該プロセッサが第1番演算の中間期間に生成するデータを一時的に記憶する。

【0008】

次に、メモリ26内における第1番結果F1FO（不図示）からの第1番結果検索後、共同プロセッサ14<sub>1</sub>はその第1番結果に対して対数関数等の第2番演算を実行する。この第2番演算は第2番結果を生み出し、それを共同プロセッサ14<sub>1</sub>がメモリ26内に規定された第2番結果F1FO（不図示）に記憶する。典型的には、共同プロセッサ14<sub>1</sub>はメモリ24<sub>1</sub>内に記憶されたプログラムを実行し、そのプログラムの制御の下で上述した動作を実行する。共同プロセッサ14<sub>1</sub>はメモリ24<sub>1</sub>を作業メモリとしても使用し得て、当該共同プロセッサが第2番演算の中間期間に生成するデータを一時的に記憶する。

10

【0009】

次に共同プロセッサ24<sub>2</sub>〜24<sub>n</sub>は、共同プロセッサ24<sub>1</sub>に対して先に議論されたものと同様に、（第2番結果〜第（n-1）番）結果に対して（第3番演算〜第n番）演算を順次実行する。

【0010】

共同プロセッサ24<sub>n</sub>によって実行される第n番演算は最終結果、即ち処理データを生み出す。共同プロセッサ24<sub>n</sub>はその処理データをメモリ26内に規定された処理データF1FO（不図示）内にロードし、遠隔装置（図1では不図示）がこのF1FOからその処理データを検索する。

20

【0011】

マスター・プロセッサ12及び共同プロセッサ14は処理アルゴリズムの種々の演算を同時に実行するので、計算マシン10は、しばしば、種々の演算を順次実行する単一プロセッサを有する計算マシンよりも生データを高速に処理することができる。詳細には、単一プロセッサは、生データから成る先行集合に対する全（n+1）個の演算を実行するまで、生データから成る新しい集合を検索できない。しかし、以上に議論したパイプライン技術を用いて、マスター・プロセッサ12は第1演算だけを実行後に生データから成る新しい集合を検索できる。結果として、所与のクロック周波数でこのパイプライン技術は、単一プロセッサ・マシン（図1では不図示）と比較して約n+1倍だけマシン10が生データを処理する速度を増大することができる。

30

【0012】

代替的には、計算マシン10は、生データに対するFFT等の処理アルゴリズムの（n+1）例を同時に実行することによって該生データを並列して処理し得る。即ち、もしそのアルゴリズムが先行する例において先に記載されたような（n+1）個の順次演算を含めば、マスター・プロセッサ12及び共同プロセッサ14の各々は生データからそれぞれが成る各集合に対して、順次、全（n+1）個の演算を実行する。その結果として、所与のクロック周波数で、先のパイプライン技術と同様のこの並列処理技術は、単一プロセッサ・マシン（図1では不図示）と比較して約n+1倍だけマシン10が生データを処理する速度を増大することができる。

40

【0013】

残念ながら、計算マシン10は単一プロセッサ・計算マシン（図1では不図示）と比べてより迅速にデータを処理できるが、マシン10のデータ処理速度はしばしばプロセッサ・クロックの周波数より非常に小さい。詳細には、計算マシン10のデータ処理速度はマスター・プロセッサ12及び共同プロセッサ14がデータ処理するのに必要な時間によって制限される。簡略化のため、この速度制限の例はマスター・プロセッサ12と連携して議論されているが、この議論は共同プロセッサ14にも適用されることを理解して頂きたい。先に議論されたように、マスター・プロセッサ12は所望の方式でデータを操作すべくプロセッサを制御するプログラムを実行する。このプログラムはプロセッサ12が実行

50

する複数の命令から成るシーケンスを含む。残念ながら、プロセッサ12は典型的には単一命令を実行するために多数のクロック・サイクルを必要とし、そしてしばしばデータの単一値を処理すべく多数の命令を実行しなければならない。例えば、プロセッサ12が第1データ値A（不図示）を第2データ値B（不図示）で乗算することを仮定する。第1クロック・サイクル中、プロセッサ12はメモリ22から乗算命令を検索する。第2及び第3クロック・サイクル中、プロセッサ12はメモリ26からA及びBをそれぞれ検索する。第4クロック・サイクル中、プロセッサ12はA及びBを乗算し、そして第5クロック・サイクル中に結果としての積をメモリ22或は26に記憶するか、或は、その結果としての積を遠隔装置（不図示）に提供する。これは最良ケースのシナリオであり、その理由は多くの場合にプロセッサ12はカウンタの初期化及び閉鎖等のオーバーヘッド・タスクに対して付加的なクロック・サイクルを必要とするからである。それ故に、よくてもプロセッサ12はA及びBを処理すべく5クロック・サイクルを必要とするか、或は、1データ値当たり平均2.5クロック・サイクルを必要とする。

#### 【0014】

結果として、計算マシン10がデータを処理する速度は、しばしば、マスター・プロセッサ12及び共同プロセッサ14を駆動するクロックの周波数より非常に低い。例えば、もしプロセッサ12は1.0ギガヘルツ（GHz）でクロックされるが、1データ値当たり平均2.5クロック・サイクルを必要とすれば、効果的なデータ処理速度は（1.0GHz）／2.5＝0.4GHzと同等である。この効果的なデータ処理速度は、しばしば、1秒当たり演算数の単位で特徴付けされる。それ故に、この例において、1.0GHzのクロック速度で、プロセッサ12は0.4ギガ演算数／秒（Gops）で使用限界が定められる。

#### 【0015】

図2は、所与クロック周波数で且つしばしば該パイプラインがクロックされる速度と略同一速度で、プロセッサが可能であるよりは高速で典型的にはデータを処理できるハードウェアに組み込まれたデータ・パイプライン30のブロック線図である。パイプライン30は、プログラム命令を実行することなく、各データに対する各演算を各々が実行する演算子回路32<sub>1</sub>～32<sub>n</sub>を含む。即ち、所望の演算は回路32内に「書き込み」が為されて、それがプログラム命令の必要性なしに自動的にその演算を具現化するように為す。プログラム命令の実行と関連されたオーバーヘッドを減ずることによって、パイプライン30は所与のクロック周波数でプロセッサが可能であるよりは単位秒当たりより多くの演算を典型的には実行する。

#### 【0016】

例えば、パイプライン30は所与のクロック周波数でプロセッサが可能であるよりは高速で以下の数式1をしばしば解くことができる。

$$Y(x_k) = (5x_k + 3) \cdot 2^{k^k}$$

ここで、 $x_k$ は複数の生データ値から成るシーケンスを表す。この例において、演算子回路32<sub>1</sub>は $5x_k$ を計算する乗算器であり、回路32<sub>2</sub>は $5x_k + 3$ を計算する加算器であり、そして回路32<sub>n</sub>（ $n=3$ ）は $(5x_k + 3) \cdot 2^{k^k}$ を計算する乗算器である。

#### 【0017】

第1クロック・サイクル $k=1$ 中、回路32<sub>1</sub>はデータ値 $x_1$ を受け取って、それを5で乗じて、 $5x_1$ を生成する。

#### 【0018】

第2クロック・サイクル $k=2$ 中、回路32<sub>2</sub>は回路32<sub>1</sub>から $5x_1$ を受け取って、3を加えて、 $5x_1 + 3$ を生成する。またこの第2クロック・サイクル中に回路32<sub>1</sub>は $5x_2$ を生成する。

#### 【0019】

第3クロック・サイクル $k=3$ 中、回路32<sub>3</sub>は回路32<sub>2</sub>から $5x_1 + 3$ を受け取って、 $2^{k^k}$ で乗じて（効果的としては、 $x_1$ だけ $5x_1 + 3$ を左シフトする）、第1結果 $(5x_1 + 3) \cdot 2^{k^k}$ を生成する。またこの第3クロック・サイクル中に回路32<sub>1</sub>は $5x_3$ を生成

し、回路 32<sub>2</sub>は  $5x_2 + 3$  を生成する。

【0020】

このようにしてパイプライン 30は、全ての生データ値が処理されるまで、引き続く生データ値  $x_i$  の処理を続行する。

【0021】

結果として、生データ値  $x_i$  の受け取り後の2つのクロック・サイクルの遅延、即ち、この遅延はパイプライン 30の待ち時間としばしば呼称され、パイプラインは結果  $(5x_i + 3)2^{k-1}$  を生成し、その後、1つの結果を生成する、即ち各クロック・サイクル毎に  $(5x_2 + 3)2^{k-2}$ 、 $(5x_3 + 3)2^{k-3}$ 、 $\dots$ 、 $(5x_n + 3)2^{k-n}$  を生成する。

【0022】

待ち時間を無視して、パイプライン 30はこうしてクロック速度と同等のデータ処理速度を有する。比較して、マスター・プロセッサ 12及び共同プロセッサ 14（図1）が先の例におけるようにクロック速度の0.4倍であるデータ処理速度を有すると仮定すれば、パイプライン 30は、所与のクロック速度で、計算マシン 10（図1）よりも2.5倍高速でデータを処理できる。

【0023】

更に図2で参照されるように、設計者はフィールド・プログラマブル・ゲート・アレイ（FPGA）等のプログラマブル・ロジックIC（PLIC）にパイプライン 30を具現化することを選ぶ可能性があり、その理由はPLICが特殊用途IC（ASIC）が為すよりも多くの設計及び変更の柔軟性を許容するからである。PLIC内にハードウェアに組み込まれた接続を構成するため、設計者はPLIC内に配置された相互接続構成レジスタを単に所定バイナリー状態に設定する。全てのこうしたバイナリー状態の組み合わせはしばしば「ファームウェア」と呼称される。典型的には、設計者はこのファームウェアをPLICと結合された不揮発性メモリ（図2では不図示）内にロードする。PLICを「ターンオン」すると、それはファームウェアをそのメモリから相互接続構成レジスタにダウンロードする。それ故に、PLICの機能を変更すべく、設計者は単にそのファームウェアを変更して、PLICがその変更されたファームウェアを相互接続構成レジスタにダウンロードすることを可能とする。ファームウェアを単に変更することによってPLICを変更する能力は、モデル作成段階中や「フィールド内」にパイプライン 30をアップグレードするために特に有用である。

【0024】

残念ながら、ハードウェアに組み込まれたパイプライン 30は重要な意思決定、特に入れ子意思決定を引き起こすアルゴリズムを実行すべき最良の選択でない可能性がある。プロセッサは、典型的には、入れ子意思決定命令（例えば、「もしAであれば、Bを為し、またもしCであれば、Dを為し、 $\dots$ 、またnを為し等々」のように、入れ子条件命令）を、比肩する長さの演算命令（例えば、「 $A+B$ 」）を実行できる程に高速に実行できる。しかしパイプライン 30は、比較的単純な決定（例えば、「 $A > B$ ？」）を効率的に為し得るが、典型的にはプロセッサができる程に効率的に入れ子決定（例えば、「もしAであれば、Bを為し、またもしCであれば、Dを為し、 $\dots$ 、またnを為す」）を為すことができない。この非効率性の1つの理由は、パイプライン 30はほんの僅かなオンボード・メモリしか持たないことがあり、したがって外部作業／プログラム・メモリ（不図示）にアクセスすることを必要とすることがあるからである。そして、こうした入れ子決定を実行すべくパイプライン 30を設計することができるが、必要とされる回路のサイズ及び複雑性はしばしばそうした設計を非現実的に為し、特にアルゴリズムが多数の種々の入れ子決定を含む場合でにそうである。

【0025】

結果として、プロセッサは典型的には重要な意思決定を必要とする用途において使用され、ハードウェアに組み込まれたパイプラインは殆ど意思決定が為されないか或は意思決定されない「ナンバークランピング（数値データ処理）」用途に典型的には限定される。

【0026】



更には、下記に議論されるように、典型的には、特にパイプライン30が多数のPLI Cを含む場合、図2のパイプライン等のハードウェアに組み込まれたパイプラインを設計／変更するよりも、図1の計算マシン10等のプロセッサに基づく計算マシンを設計／変更することが非常に易しい。

#### 【0027】

プロセッサ及びそれらの周辺機器（例えば、メモリ）等の計算構成要素は、典型的には、プロセッサに基づく計算マシンを形成すべくそれら構成要素の相互接続を補助する工業規格通信インターフェースを含む。

#### 【0028】

典型的には、規格通信インターフェースは2つの層、即ち、物理層及びサービス層を含む。

#### 【0029】

物理層は、回路とこの回路のインターフェース及び動作パラメータを形成する対応回路相互接続とを含む。例えば、物理層はそれら構成要素を1つのバスに接続するピンと、それらのピンから受け取ったデータをラッチするバッファと、信号をそれらピンに駆動するドライバとを含む。動作パラメータは、ピンが受け取るデータ信号の許容可能電圧範囲と、データの書き込み及び読み取りのための信号タイミングと、動作の支援されたモード（例えば、バーストモード、ページモード）とを含む。従来の物理層はトランジスタ・ラジスタ論理（TTL）及びRAMBUSを含む。

#### 【0030】

サービス層は、計算構成要素のデータ転送のためのプロトコルを含む。このプロトコルはデータのフォーマットと、構成要素によるフォーマット済みデータの送受信の方式とを含む。従来の通信プロトコルは、ファイル転送プロトコル（FTP）及び伝送制御プロトコル／インターネット・プロトコル（TCP/IP）を含む。

#### 【0031】

結果として、製造業者やその他は工業規格通信インターフェースを有する計算構成要素を典型的には設定するので、そうした構成要素のインターフェースを典型的には設計でき、それを他の計算構成要素と比較的少ない労力で相互接続することができる。これは、計算マシンの他の部分の設計に設計者自信の時間を殆ど費やすことを可能として、各種構成要素を追加或は除去することによってそのマシンを変更することを可能としている。

#### 【0032】

工業規格通信インターフェースを支援する計算構成要素を設計することは、設計ライブラリから既存の物理層を用いることによって設計時間を節約することを可能としている。これは、設計者が構成要素を既製の計算構成要素と容易にインターフェースすることを保証するものである。

#### 【0033】

そして、共通した工業規格通信インターフェースを支援する計算構成要素を用いる計算マシンを設計することは、設計者がそれら構成要素を少しの時間及び労力で相互接続することを可能としている。それら構成要素は共通インターフェースを支援するので、設計者はそれらをシステム・バスを介して少しの設計労力で相互接続することができる。そして、その支援されたインターフェースは工業規格であるので、マシンを容易に変更することができる。例えば、システム設計が進化するに伴って種々の構成要素及び周辺機器をマシンに追加することができるか、或は、テクノロジーが進化するに伴って次世代の構成要素を追加／設計することが可能である。更には、構成要素が通常の工業規格サービス層を支援するので、計算マシンのソフトウェアに対応するプロトコルを具現化する既存のソフトウェア・モジュールを組み込むことができる。それ故に、インターフェース設計が本質的には既に整っているのので少しの労力で構成要素をインターフェースでき、よって、マシンに所望の機能を実行させるマシンの各種部分（例えばソフトウェア）の設計に集中することができる。

#### 【0034】



しかし残念ながら、図2のパイプライン30等のハードウェアに組み込まれたパイプラインを形成すべく、使用されるPLIC等の各種構成要素に対する既知の工業規格サービス層が全くない。

#### 【0035】

結果として、多数のPLICを有するパイプラインを設計すべく、多大な時間を費やし、「ゼロから」種々のPLICの間の通信インターフェースのサービス層を設計し且つデバッグする多大な労力を行行使する。典型的には、そうしたその場限りのサービス層は種々のPLIC間で転送されるデータのパラメータに依存する。同じように、プロセッサとインターフェースするパイプラインを設計すべく、ゼロからのパイプライン及びプロセッサの間の通信インターフェースのサービス層の設計及びデバッグに関して多大な時間を費やし且つ多大な労力を行行使する必要がある。

10

#### 【0036】

同様に、そうしたパイプラインをPLICを該パイプラインに追加することによって変更すべく、典型的には、その追加されたPLICと既存のPLICとの間の通信インターフェースのサービス層の設計及びデバッグに関して多大な時間を費やし且つ多大な労力を行行使する。同じように、プロセッサを追加することによってパイプラインを変更すべく、或は、パイプラインを追加することによって計算マシンを変更すべく、パイプライン及びプロセッサの間の通信インターフェースのサービス層の設計及びデバッグに関して多大な時間を費やし且つ多大な労力を行行使しなければならないであろう。

#### 【0037】

20

結果として、図1及び図2で参照されるように、多数のPLICをインターフェースすることとプロセッサをパイプラインにインターフェースすることとの難しさのため、計算マシンを設計する際に多大な妥協を為すことがしばしば強えられる。例えば、プロセッサに基づく計算マシンでは、ナンバークランピング速度を、複雑な意思決定を為す能力に対する設計/変更の柔軟性と交換することを強えられる。逆に、ハードウェアに組み込まれたパイプラインに基づく計算マシンでは、複雑な意思決定を為す能力と設計/変更の柔軟性を、ナンバークランピング速度と交換することを強えられる。更には、多数のPLICをインターフェースすることに関する難しさのため、少数のPLICよりも多くのPLICを有するパイプラインに基づくマシンを設計することはしばしば実際的ではない。その結果、実際的なパイプラインに基づくマシンはしばしば制限された機能しか有さない。そして、プロセッサをPLICとインターフェースすることに関する難しさのため、プロセッサを1つのPLICより多くのPLICにインターフェースすることは実際的ではない。その結果、プロセッサ及びパイプラインを組み合わせることによって獲得される利益は最少となる。

30

#### 【発明の開示】

#### 【発明が解決しようとする課題】

#### 【0038】

それ故に、プロセッサに基づくマシンの意思決定を為す能力を、ハードウェアに組み込まれたパイプラインに基づくマシンのナンバークランピング速度と組み合わせることを可能とする新しい計算アーキテクチャに対する要望が生じてきている。

40

#### 【課題を解決するための手段】

#### 【0039】

本発明の実施例に従えば、パイプライン加速器はメモリと該メモリと結合されたハードウェアに組み込まれたパイプライン回路とを含む。ハードウェアに組み込まれたパイプライン回路は、データを受信し、該データをメモリにロードし、該メモリからのそのデータを検索し、その検索データを処理して、その処理されたデータを外部ソースに提供するように動作できる。

#### 【0040】

本発明の別の実施例に従えば、ハードウェアに組み込まれたパイプライン回路は、データを受信し、その受信されたデータを処理し、その処理されたデータをメモリにロードし

50

、メモリからその処理されたデータを検索して、その検索された処理データを外部ソースに提供するように動作できる。

#### 【0041】

パイプライン加速器がピア-ベクトル・マシンの一部としてのプロセッサと結合されている場合、メモリはハードウェアに組み込まれたパイプライン回路とプロセッサが実行するアプリケーションとの間でのデータの転送（単向性であろうが二方向性であろうが）を補助する。

#### 【発明を実施するための最良の形態】

#### 【0042】

図3は、本発明の一実施例に従ったピア-ベクトル・アーキテクチャを有する計算マシン40の概略ブロック線図である。ホストプロセッサ42に加えて、ピア-ベクトル・マシン40はパイプライン加速器44を含み、それがデータ処理の少なくとも一部を実行して、図1の計算マシン10における共同プロセッサ14の列と効果的に置き換わる。それ故に、ホストプロセッサ42及び加速器44（又は以下に議論されるようにそのユニット）はデータ・ベクトルを前後に転送できる「ピア」である。加速器44はプログラム命令を実行しないので、所与のクロック周波数で共同プロセッサの列ができるものよりも著しく高速にデータに対して数学的に集中的な演算を典型的には実行する。結果として、プロセッサ42の意思決定能力と加速器44のナンバーランチング能力とを組み合わせることによって、マシン40はマシン10等の従来の計算マシンと同一の能力を有するが、しばしばそれよりもデータをより高速に処理することができる。更には、以下に議論されるように、加速器44にホストプロセッサ42の通信インターフェースと互換性がある通信インターフェースを設けることが、特にプロセッサの通信インターフェースが工業規格である場合に、マシン40の設計及び変更を補助する。そして、加速器44が多数のパイプライン・ユニット（例えば、PLICに基づく回路）を含む場合、それら各ユニットに同一の通信インターフェースを設けることが、特にそれら通信インターフェースが工業規格インターフェースと互換性がある場合に、当該加速器の設計及び変更を補助する。更には、マシン40は以下に議論されると共に先行して引用された特許出願におけるような他の長所等をも提供し得る。

#### 【0043】

更に図3で参照されるように、ホストプロセッサ42及びパイプライン加速器44に加えて、ピア-ベクトル・計算マシン40は、プロセッサ・メモリ46、インターフェース・メモリ48、バス50、ファームウェア・メモリ52、任意選択的な生データ入力ポート54、処理済みデータ出力ポート58、並びに、任意選択的なルータ61を含む。

#### 【0044】

ホストプロセッサ42は処理ユニット62及びメッセージ・ハンドラー64を含み、プロセッサ・メモリ46は処理ユニット・メモリ66及びハンドラー・メモリ68を含み、そのそれぞれがプロセッサ・ユニット及びメッセージ・ハンドラーに対するプログラム及び作業の両メモリとして役立っている。プロセッサ・メモリ46は、加速器コンフィギュレーション・レジストリ70及びメッセージ・コンフィギュレーション・レジストリ72をも含み、それらが、ホストプロセッサ42が加速器44の機能を構成すると共に、該ホストプロセッサ42がメッセージ・ハンドラー64が送信及び受信するメッセージのフォーマットを構成することを可能とするそれぞれのコンフィギュレーション・データを記憶する。

#### 【0045】

パイプライン加速器44は少なくとも1つのPLIC（不図示）上に配置され、プログラム命令を実行することなしに各データを処理するハードウェアに組み込まれたパイプライン74、74'を含む。ファームウェア・メモリ52は加速器44に対するコンフィギュレーション・ファームウェアを記憶する。もし加速器44が多数のPLIC上に配置されたら、それらPLIC及びそれらの各ファームウェア・メモリは多数パイプライン・ユニット内に配置され得る（図4）。加速器44及びパイプライン・ユニットは、以下に議

論されると共に、先行して引用された「多数パイプライン・ユニットを有するパイプライン加速器、関連計算マシン、並びに、方法」と題された特許文献5に更に議論されている。代替的には、加速器44は少なくとも1つのASIC上に配置され得て、よって構成不可能な内部相互接続を有し得る。この代替例において、マシン40はファームウェア・メモリ52を省略し得る。更には、加速器44が多数パイプライン74を含んで示されているが、ただ1つのパイプラインを含み得る。加えて、図示されていないが、加速器44はデジタル信号プロセッサ(DSP)等の1つ或はそれ以上のプロセッサを含み得る。更には、図示されていないが、加速器44はデータ入力ポート及び/或はデータ出力ポートを含み得る。

【0046】

10

ピア-ベクトル・マシン40の一般動作は、先行して引用された「改善された計算アーキテクチャ、関連システム、並びに、方法」と題された特許文献2に議論されており、パイプライン加速器44の構造及び動作は図4乃至図9と連携された以下に議論されている。

【0047】

図4は、本発明の一実施例に従った図3のパイプライン加速器44の概略ブロック線図である。

【0048】

加速器44は1つ或はそれ以上のそうしたパイプライン・ユニット78を含み、それらの各々はPLIC或はASIC等のパイプライン回路80を含む。以下で更に議論されると共に先行して引用された「多数パイプライン・ユニットを有するパイプライン加速器、関連計算マシン、並びに、方法」と題された特許文献5にあるように、各パイプライン・ユニット78はホストプロセッサ42の「ピア」であると共に加速器44の他のパイプライン・ユニットの「ピア」である。即ち、各パイプライン・ユニット78はホストプロセッサ42或は他の任意のパイプライン・ユニットと直接通信できる。よって、このピア-ベクトル・アーキテクチャは、もしパイプライン・ユニット78の全てがマスターパイプライン・ユニット(不図示)或はホストプロセッサ42等の中央箇所を通じて通信した場合に生ずることとなるデータ「ボトルネック」を防止する。更には、これは、マシンに対する重大な変更なしに、ピアを追加するか或はピア-ベクトル・マシン40(図3)からピアを除去することを可能とする。

20

30

【0049】

パイプライン回路80は通信インターフェース82を含み、それが、ホストプロセッサ42(図3)等のピアと、通信シュール84を介したハードウェアに組み込まれたパイプライン74、74。(図3)コントローラ86、例外マネージャ88、並びに、コンフィギュレーション・マネージャ90等の、パイプライン回路の他の構成要素との間でデータを転送する。パイプライン回路80は工業規格バス・インターフェース91をも含み得る。代替的には、インターフェース91の機能は通信インターフェース82内に含まれ得る。

【0050】

パイプライン回路80の複数の構成要素を複数の個別モジュールとして設計することによって、そのパイプライン回路の設計をしばしば簡略化することができる。即ち、それら構成要素の各々を個別に設計及び試験することができ、次いでそれらを統合するものであり、それはソフトウェア或はプロセッサに基づく計算システム(図1のシステム10等)を設計する際に行われることと非常に似ている。加えて、これら構成要素、特に他のパイプライン設計においてたぶん頻繁に使用されるような通信インターフェース82等の構成要素を規定するハードウェア記述言語(HDL)をライブラリ(不図示)内に記憶でき、よって同一構成要素を使用する将来のパイプライン設計の設計及び試験の時間を低減する。即ち、ライブラリからHDLを使用することによって、設計者はスクラッチから先行して具現化された構成要素を再設計する必要性がなく、よって設計者の努力を先行して具現化されていない構成要素の設計に対して、或は、先行して具現化された構成要素の変更に対して集中できる。更には、ライブラリ内にパイプライン回路80或はパイプライン加速

40

50

器 4 4 の多数バージョンを画成する HDL を記憶できて、既存の設計の中から精選及び選択できるように為す。

#### 【0051】

通信インターフェース 8 2 はメッセージ・ハンドラー 6 4 (図 3) によって認識されるフォーマットでデータを送受信し、よってピア・ベクトル・マシン 4 0 (図 3) の設計及び変更を典型的には補助する。例えば、もしデータ・フォーマットが高速 1/0 フォーマット等の工業規格であれば、ホストプロセッサ 4 2 及び加速器 4 4 の間にカスタムインターフェースを設計する必要がない。更には、パイプライン回路 8 0 に非バス・インターフェースの代わりにパイプライン・バス 5 0 を介してホストプロセッサ 4 2 (図 3) 等の他のピアと通信させることを可能にすることによって、パイプライン・ユニットが追加或は除去されるたびにゼロから非バス・インターフェースを再設計する代わりに、パイプライン・バスにそれらパイプライン・ユニット (又はそれらを保持する回路カード) を単に接続或は接続解除することによってパイプライン・ユニット 7 8 の数を変更できる。

#### 【0052】

ハードウェアに組み込まれたパイプライン 7 4<sub>1</sub> - 7 4<sub>n</sub> は、図 3 と連携されて先に議論されると共に、先行して引用された「改善された計算アーキテクチャ、関連システム、並びに、方法」と題された特許文献 2 に議論されているようにデータに対して各演算を実行し、通信シェル 8 4 はそれらパイプラインをパイプライン回路 8 0 の他の構成要素や該パイプライン回路外部の回路 (以下に議論されるデータ・メモリ 9 2 等) とインターフェースする。

#### 【0053】

コントローラ 8 6 はハードウェアに組み込まれたパイプライン 7 4<sub>1</sub> - 7 4<sub>n</sub> を同期し、通信に応じて、即ち他のピアからの「事象」に応じて、それらが各データ演算を実行するシーケンスをモニタし且つ制御する。例えば、ホストプロセッサ 4 2 等のピアはパイプライン・バス 5 0 を介してパイプライン・ユニット 7 8 に事象を送信し得て、そのピアがデータから成るブロックをパイプライン・ユニットに送信し終えたことを示して、ハードウェアに組み込まれたパイプライン 7 4<sub>1</sub> - 7 4<sub>n</sub> にそのデータの処理を始めさせる。データを含む事象は典型的にはメッセージと呼称され、データを含まない事象は典型的には「ドアベル」と呼称される。更には、図 5 と連携して以下に議論されるように、パイプライン・ユニット 7 8 も同期信号に応じてパイプライン 7 4<sub>1</sub> - 7 4<sub>n</sub> を同期し得る。

#### 【0054】

例外マネージャ 8 8 はハードウェアに組み込まれたパイプライン 7 4<sub>1</sub> - 7 4<sub>n</sub>、通信インターフェース 8 2、通信シェル 8 4、コントローラ 8 6、並びに、バス・インターフェース 9 1 の状況をモニタし、ホストプロセッサ 4 2 (図 3) に例外を報告する。例えば、もし通信インターフェース 8 2 におけるバッファがオーバーフローすれば、例外マネージャ 8 8 はこれをホストプロセッサ 4 2 に報告する。例外マネージャはその例外を生んだ問題を修正するか或はその修正を試みることも可能である。例えば、オーバーフローしているバッファに対して例外マネージャ 8 8 は、直接的或は以下に議論されるようなコンフィギュレーション・マネージャ 9 0 を介して、そのバッファのサイズを増大し得る。

#### 【0055】

コンフィギュレーション・マネージャ 9 0 はハードウェアに組み込まれたパイプライン 7 4<sub>1</sub> - 7 4<sub>n</sub>、通信インターフェース 8 2、通信シェル 8 4、コントローラ 8 6、例外マネージャ 8 8、並びに、インターフェース 9 1 のソフト・コンフィギュレーションを、ホストプロセッサ 4 2 (図 3) からのソフト・コンフィギュレーション・データに応じて設定し、これは先に引用された「改善された計算アーキテクチャ、関連システム、並びに、方法」と題された特許文献 2 に議論され、ハード・コンフィギュレーションはパイプライン回路 8 0 のトランジスタ及び回路ブロックのレベル上における実際のトポロジーを示し、ソフト・コンフィギュレーションはハード構成された構成要素の物理的パラメータ (例えば、データ幅、テーブル・サイズ) を示す。即ち、ソフト・コンフィギュレーション・データはプロセッサ (図 4 に不図示) のレジスタにロードされ得るプロセッサの動作モー

ド（例えば、バースト・メモリ・モード）を設定するデータと同様である。例えばホストプロセッサ42は、コンフィギュレーション・マネージャ90に通信インターフェース82におけるキューの数及び各優先レベルを設定させるソフト・コンフィギュレーション・データを送信し得る。例外マネージャ88は、コンフィギュレーション・マネージャ90に、例えば、通信インターフェース82におけるオーバーフローしているバッファのサイズを増大させるソフト・コンフィギュレーション・データをも送信し得る。

【0056】

更に図4で参照されるように、パイプライン回路80に加えて、加速器44のパイプライン・ユニット78は、データ・メモリ92、任意選択的な通信バス94、並びに、パイプライン回路がPLICである場合におけるファームウェア・メモリ52（図3）を含む。

【0057】

データ・メモリ92は、ホストプロセッサ42（図3）等の別のピアとハードウェアに組み込まれたパイプライン74<sub>1</sub>〜74<sub>n</sub>との間を流れているデータをバッファすると共に、ハードウェアに組み込まれたパイプラインに対する作業メモリでもある。通信インターフェース82はデータ・メモリ92をパイプライン・バス50（通信バス94ともしあれば工業規格インターフェース91とを介して）にインターフェースし、通信シエル84はデータ・メモリをハードウェアに組み込まれたパイプライン74<sub>1</sub>〜74<sub>n</sub>にインターフェースする。

【0058】

工業規格バス・インターフェース91は、通信インターフェース82から幾つかのインターフェース回路を効果的にオフロードすることによって、通信インターフェース82のサイズ及び複雑性を低減する従来のバス・インターフェース回路である。それ故に、もしパイプライン・バス50或はルータ61（図3）のパラメータを変更することを望めば、インターフェース91を変更するだけでよく、通信インターフェース82を変更する必要がない。代替的には、パイプライン回路80の外部であるIC（不図示）内にインターフェース91を配置し得る。パイプライン回路80からインターフェース91をオフロードすることは、例えばハードウェアに組み込まれたパイプライン74<sub>1</sub>〜74<sub>n</sub>及びコントローラ86の用途のパイプライン回路上のリソースを解放する。或は、先に議論されたように、バス・インターフェース91は通信インターフェース82の一部であり得る。

【0059】

パイプライン回路80がPLICである図3と連携されて先に議論されたように、ファームウェア・メモリ52はパイプライン回路のハード・コンフィギュレーションを設定するファームウェアを記憶する。このメモリ52はファームウェアを加速器44の構成中にパイプライン回路80にロードし、加速器の構成中或はその後に通信インターフェース82を介してホストプロセッサ42（図3）から変更されたファームウェアを受信し得る。ファームウェアのローディング及び受信は、先行して引用された「プログラマブル回路、関連計算マシン、並びに、方法」と題された特許文献4に更に議論されている。

【0060】

図4で更に参照されるように、パイプライン回路80、データ・メモリ92、並びに、ファームウェア・メモリ52は回路ボード或はカード98上に配置され得て、パーソナルコンピュータ（不図示）におけるドーターカードがマザーボードのスロットにプラグインされ得ることと非常に類似して、パイプライン・バス・コネクタ（不図示）にプラグインされ得る。図示されていないが、従来のIC、電力調整器等の構成要素、並びに、電力シークンサも周知のようにカード98上に配置され得る。

【0061】

パイプライン・ユニット78の構造及び動作の更なる詳細は図5と連携されて以下に議論される。

【0062】

図5は、本発明の実施例に従った図4のパイプライン・ユニット78のブロック線図で

ある。簡略化のため、ファームウェア・メモリ52は図5から省略されている。パイプライン回路80はマスターCLOCK信号を受信し、それが直接的或は間接的にパイプライン回路の以下に記載される構成要素を駆動する。パイプライン回路80は従来方式でマスターCLOCK信号から1つ或はそれ以上のスレーブCLOCK信号（不図示）を生成し得る。またパイプライン回路80は以下に議論されるように同期信号SYNCを受信し得る。

#### 【0063】

データ・メモリ92は入力デュアル-ポート-スタティック-ランダム-アクセス・メモリ（DPSRAM）100、出力DPSRAM102、並びに、任意選択的な作業DPSRAM104を含む。

#### 【0064】

入力DPSRAM100は、通信インターフェース82を介して、ホストプロセッサ42（図3）等のピアからデータを受信するための入力ポート106を含むと共に、通信シェル84を介してそのデータをハードウェアに組み込まれたパイプライン74<sub>1</sub>〜74<sub>n</sub>に提供するための出力ポート108を含む。一方がデータ入力のためであり且つ他方がデータ出力のためであるこれら2つのポートを有することは、DPSRAM100へのデータ転送／DPSRAM100からのデータ転送の速度及び効率を増大するが、その理由としては通信インターフェース82がDPSRAMにデータを書き込むことができる一方でパイプライン74<sub>1</sub>〜74<sub>n</sub>がそのDPSRAMからデータを読み取ることができるからである。更に、先に議論されたように、ホストプロセッサ42等のピアからデータをバッファすべくDPSRAM100を用いることは、そのピア及びパイプライン74<sub>1</sub>〜74<sub>n</sub>に相互に対して非同期的に動作させることを可能としている。即ちピアは、パイプライン74<sub>1</sub>〜74<sub>n</sub>が現行動作を完了するのを「待機」することなしに、データをパイプラインに送信できる。同様に、パイプライン74<sub>1</sub>〜74<sub>n</sub>はピアがデータ送信動作を完了するのを「待機」することなしにデータを検索できる。

#### 【0065】

同じように、出力DPSRAM102は、通信シェル84を介して、ハードウェアに組み込まれたパイプライン74<sub>1</sub>〜74<sub>n</sub>からデータを受信するための入力ポート110を含むと共に、そのデータを通信インターフェース82を介してホストプロセッサ42（図3）等のピアに提供するための出力ポート112を含む。先に議論されたように、これら2つのデータ・ポート110（入力）及び112（出力）はDPSRAM102へのデータ転送／DPSRAM102からのデータ転送の速度及び効率を増大し、パイプライン74<sub>1</sub>〜74<sub>n</sub>からデータをバッファすべくDPSRAM102を用いることは、ピア及びパイプラインに相互に対して非同期的に動作させることを可能としている。即ち、パイプライン74<sub>1</sub>〜74<sub>n</sub>は、出力データ・ハンドラー126がそのピア或は別のピアにデータ転送を完了するのを「待機」することなしにピアにデータを発行できる。同様に、出力データ・ハンドラー126は、パイプライン74<sub>1</sub>〜74<sub>n</sub>がデータ発行動作を完了するのを「待機」することなしにデータをピアに転送できる。

#### 【0066】

作業DPSRAM104は、通信シェル84を介して、ハードウェアに組み込まれたパイプライン74<sub>1</sub>〜74<sub>n</sub>からデータを受信するための入力ポート114を含むと共に、その通信シェルを介してそのデータをパイプラインに戻すための出力ポート116を含む。DPSRAM100から受信された入力データを処理している間に、パイプライン74<sub>1</sub>〜74<sub>n</sub>は部分的に処理された、即ち中間データをそのデータの処理を続行する前に一時的に記憶する必要があり得る。例えば、パイプライン74<sub>1</sub>等の第1パイプラインはパイプライン74<sub>2</sub>等の第2のパイプラインによる更なる処理のために中間データを生成し得て、よって、第1パイプラインは第2パイプラインがその中間データを検索するまでそれを一時的に記憶する必要があり得る。作業DPSRAM104はこの一時的記憶装置を提供する。先に議論されたように、2つのデータ・ポート114（入力）及び116（出力）はパイプライン74<sub>1</sub>〜74<sub>n</sub>及びDPSRAM104の間でのデータ転送の速度及び効



率を増大する。更には別個の作業DPSRAM104を含むことは、DPSRAM100及び102にデータ入力バッファ及びデータ出力バッファ専用としてそれぞれ機能させることによってパイプライン回路80の速度及び効率を典型的には増大する。しかし、パイプライン回路80に対する僅かな変更によって、DPSRAM100及び102の両方又は片方が、DPSRAM104が省略された際に、そしてそれが存在したとしても、パイプライン74、74aに対する作業メモリでもあり得る。

#### 【0067】

DPSRAM100、102、104はパイプライン回路80の外部であるとして記載されているが、それらDPSRAM若しくはそれと同等物の内の1つ或はそれ以上をそのパイプライン回路の内部とすることが可能である。

10

#### 【0068】

更に図5で参照されるように、通信インターフェース82は、工業規格バス・アダプタ118、入力データ・ハンドラー120、入力データ及び入力事象キュー122及び124、出力データ・ハンドラー126、並びに、出力データ及び出力事象キュー128及び130を含む。それらのキュー122、124、128、130がそれぞれ単一キューとして示されているが、それらキューの内の1つ或はそれ以上は、例えば、それらキューに記憶された値の優先順位、或は、それら値が表す各データの優先順位による分離を可能とするサブ・キュー（不図示）を含み得る。

#### 【0069】

工業規格バス・アダプタ118は、通信バス94を介して、パイプライン回路80及びパイプライン・バス50（図4）の間でのデータの転送を可能とする物理層を含む。それ故に、もしバス94のパラメータの変更が望まれたならば、アダプタ118だけを変更する必要がある、通信インターフェース82全体を変更する必要性はない。工業規格バス・インターフェース91がパイプライン・ユニット78から省略されている場合、アダプタ118をパイプライン・バス50及びパイプライン回路80の間での直接的なデータ転送を可能とするように変更し得る。この後者の具現化例において、変更アダプタ118はバス・インターフェース91の機能を含み、バス50のパラメータの変更が望まれたならばアダプタ118を変更することだけが必要である。

20

#### 【0070】

入力データ・ハンドラー120は工業規格アダプタ118からデータを受信し、そのデータを入力ポート106を介してDPSRAM100にロードし、そのデータを指すポインタと対応するデータ識別子を生成してそれらを入力データ・キュー122に記憶する。もしそのデータがホストプロセッサ42（図3）等のピアからのメッセージのペイロードであれば、入力データ・ハンドラー120はそのデータをDPSRAM100にロードする前にメッセージからそれを抽出する。入力データ・ハンドラー120はインターフェース132を含み、それがDPSRAM100の入力ポート106にデータを書き込むが、それは図6と連携された以下に更に議論される。代替的には、入力データ・ハンドラー120は抽出ステップを省略できると共にメッセージ全体をDPSRAM100にロードできる。

30

#### 【0071】

また入力データ・ハンドラー120は工業規格アダプタ118から事象を受信して、その事象を入力事象キュー124にロードする。

40

#### 【0072】

更には、入力データ・ハンドラー120は有効マネージャ134を含み、それが受信されたデータ或は事象がパイプライン回路80に対して意図されているかを決定する。有効マネージャ134は、データ或は事象を含むメッセージのヘッダー（或はその一部）を分析することによって、データ或は事象のタイプを分析することによって、或は、データ或は事象の例証識別（即ちデータ／事象が意図されているハードウェアに組み込まれたパイプライン74）を分析することによって、先の決定を為し得る。もし入力データ・ハンドラー120がパイプライン回路80に対して意図されていないデータ或は事象を受信すれ

50

ば、有効マネージャ134は入力データ・ハンドラーがその受信データ／事象をロードすることを禁止する。ピア・ベクトル・マシン40がルータ61（図3）を含んで、パイプライン・ユニット78が該パイプライン・ユニットに対して意図されているデータ／事象のみを受信するように為す場合、有効マネージャ134も入力データ・ハンドラー120に例外（誤って受信されたデータ／事象）やその例外を引き起こしたピアを識別する例外メッセージをホストプロセッサ42（図3）に送信させ得る。

#### 【0073】

出力データ・ハンドラー126は出力データ・キュー128によって指されたDPSRAM102の複数箇所から処理データを検索し、それら処理データを工業規格バス・アダプタ118を介してホストプロセッサ42（図3）等の1つ或はそれ以上のピアに送信する。出力データ・ハンドラー126はインターフェース136を含み、それがポート112を介してDPSRAM102から処理データを読み取る。このインターフェース136は図7と連携されて以下に更に議論される。

#### 【0074】

出力データ・ハンドラー126もパイプライン74<sub>1</sub>〜74<sub>n</sub>によって生成された事象を出力事象キュー130から検索し、工業規格バス・アダプタ118を介してホストプロセッサ42（図3）等の1つ或はそれ以上のピアにその検索された事象を送信する。

#### 【0075】

更には、出力データ・ハンドラー126は加入マネージャ138を含み、それが処理データや事象に加入しているホストプロセッサ42（図3）等のピアのリストを含み、出力データ・ハンドラーはそのリストを用いてデータ／事象を正しいピアに送信する。もしピアがデータ／事象がメッセージのペイロードであることを好めば、出力データ・ハンドラー126は加入マネージャ138からピアのネットワーク或はバス・ポート・アドレスを検索し、アドレスを含むヘッダーを生成し、データ／事象及びヘッダーからメッセージを生成する。

#### 【0076】

DPSRAM100及び102にデータを記憶し記憶されたデータを検索するための技術はポインタやデータ識別子の使用を含むが、他のデータ管理技術を実現化すべく入力及び出力データ・ハンドラー120及び126を変更することができる。そうしたデータ管理技術の従来例は、キー或はトークンを用いるポインタ、入力／出力制御（I/O C）ブロック、並びに、スプーリングを含む。

#### 【0077】

通信シェル84はハードウェアに組み込まれたパイプライン74<sub>1</sub>〜74<sub>n</sub>を出力データ・キュー128、コントローラ86、並びに、DPSRAM100、102、104にインターフェースする物理層を含む。このシェル84はインターフェース140及び142と任意選択的なインターフェース144及び146を含む。インターフェース140及び146はインターフェース136と同様であり得て、インターフェース140はポート108を介してDPSRAM100から入力データを読み取り、そしてインターフェース146はポート116を介してDPSRAM104から中間データを読み取る。インターフェース142及び144はインターフェース132と同様であり得て、インターフェース142はポート110を介してDPSRAM102に処理データを書き込み、そしてインターフェース144はポート114を介してDPSRAM104に中間データを書き込む。

#### 【0078】

コントローラ86はシーケンス・マネージャ148及び同期インターフェース150を含み、該同期インターフェースは1つ或はそれ以上の同期信号SYNCを受信する。ホストプロセッサ42（図3）等のピア、或は、ピア・ベクトル・マシン40（図3）の外部の装置（不図示）は、SYNC信号を生成し得て、以下に議論されると共に、先行して引用された「多数パイプライン・ユニットを有するパイプライン加速器、関連計算マシン、並びに、方法」と題された特許文献5に議論されるように、それがシーケンス・マネージ



ャ148をトリガーしてハードウェアに組み込まれたパイプライン74<sub>1</sub>〜74<sub>n</sub>を起動する。同期インターフェース150もSYNC信号を生成し得て、パイプライン回路80をトリガーするか或は別のピアをトリガーする。加えて、入力事象キュー124からの事象もシーケンス・マネージャ148をトリガーして、以下に議論されるように、ハードウェアに組み込まれたパイプライン74<sub>1</sub>〜74<sub>n</sub>を起動する。

#### 【0079】

シーケンス・マネージャ148は通信シェル84を介してハードウェアに組み込まれたパイプライン74<sub>1</sub>〜74<sub>n</sub>をそれらの各動作を通じて順序付けする。典型的には、各パイプライン74は、事前処理、処理、並びに、事後処理の少なくとも3つの状態を有する。事前処理中、パイプライン74は、例えば、そのレジスタを初期化して、DP SRAM100から入力データを検索する。処理中、パイプライン74は、例えば、その検索されたデータに対して演算を為し、中間データをDP SRAM104に一時的に記憶し、DP SRAM104からその中間データを検索してから、その中間データに対して演算を為して結果データを生成する。事後処理中、パイプライン74は、例えば、その結果としてのデータをDP SRAM102にロードする。それ故に、シーケンス・マネージャ148はパイプライン74<sub>1</sub>〜74<sub>n</sub>の動作又は演算をモニタして、各パイプラインにその動作状態の各々をいつ始めるべきかを命令する。そして、パイプライン・タスクを先に記載したものと異なるように各種動作状態の間に分配し得る。例えば、パイプライン74は事前処理状態中の代わりに処理状態中にDP SRAM100から入力データを検索し得る。

#### 【0080】

更には、シーケンス・マネージャ148はハードウェアに組み込まれたパイプライン74<sub>1</sub>〜74<sub>n</sub>間で所定の内部動作同期を維持する。例えば、パイプライン74<sub>1</sub>〜74<sub>n</sub>の全てがDP SRAM100からデータを同時に検索することを回避すべく、第1パイプライン74<sub>1</sub>が事前処理状態である一方で、第2パイプライン74<sub>2</sub>が処理状態、第3パイプライン74<sub>3</sub>が事後処理状態となるようにそれらパイプラインを同期させることが望ましい場合がある。1つのパイプライン74の状態が別のパイプラインの同時発生的に実行されている状態とは異なる数のクロック・サイクルを要求し得るので、パイプライン74<sub>1</sub>〜74<sub>n</sub>はもし自由に走らされると同期性を喪失し得る。結果として、特定時に、例えば多数のパイプライン74がDP SRAM100からデータを同時に検索しようとするような「ボトルネック」があり得る。同時性の喪失やその望ましくない結果を防止すべく、シーケンス・マネージャ148は、それらパイプラインの任意のものに対して次の動作状態に進めさせる前に、パイプライン74の全てに現行の動作状態を完了させる。それ故に、シーケンス・マネージャ148が現行の動作状態に対して振り分ける時間は、最も緩慢なパイプライン74にその状態を完了させるに十分な長さである。代替的には、ハードウェアに組み込まれたパイプライン74<sub>1</sub>〜74<sub>n</sub>の間で所定動作同期性を維持するための回路（不図示）はパイプライン自体内に含ませ得る。

#### 【0081】

ハードウェアに組み込まれたパイプライン74<sub>1</sub>〜74<sub>n</sub>を順序づけすると共に内部的に同期化することに加えて、シーケンス・マネージャ148はパイプラインの動作を、1つ或はそれ以上のSYNC信号、或は、入力事象キュー124内の事象に応じて、ホストプロセッサ42（図3）等の他のピアの動作、及び他の外部装置の動作と同期する。

#### 【0082】

典型的には、SYNC信号はタイムクリティカルな機能をトリガーするが著しいハードウェア・リソースを必要とし、比較して、典型的には事象は非タイムクリティカルな機能をトリガーするが著しくより少ないハードウェア・リソースを必要とする。先行して引用された「多数パイプライン・ユニットを有するパイプライン加速器、関連計算マシン、並びに、方法」と題された特許文献5に議論されているように、SYNC信号はピアからピアに直接的に経路指定されているので、例えば、パイプライン・バス50（図3）、入力データ・ハンドラー120、並びに、入力事象キュー124を通じてその道を作成しなければならない事象よりも迅速に機能をトリガーできる。しかし、それらが個別に経路指定

されるので、SYNC信号は、ルーティング・ライン、バッファ、並びに、SYNCインターフェース150等のパイプライン回路80の専用回路を必要とする。逆に、それらが既存のデータ転送下部組織（例えば、パイプライン・バス50及び入力データ・ハンドラー120）を使用するので、事象は専用の入力事象キュー124のみを必要とする。結果として、設計者はタイムクリティカルな機能のほとんど全てをトリガーすべく事象を使用する傾向がある。

#### 【0083】

以下は機能トリガーリングの一例である。ソナー・センサ要素（不図示）がパイプライン・ユニット78にデータから成るブロックを送信することを仮定すると、入力データ・ハンドラー120はこのデータをDP SRAM100に記憶し、パイプライン74はこのデータをそのDP SRAM100からDP SRAM104に転送し、そして、トリガーされると、パイプライン74はDP SRAM104からそのデータを検索して処理する。もしパイプライン74がそのデータに実行する処理がタイムクリティカルであれば、パイプライン74がDP SRAM104にデータ・ブロックの全体をローディングし終わるとすぐに、センサ要素はSYNCパルスを生成してパイプライン74をインターフェース150及びシーケンス・マネージャ148を介してトリガーする。パイプライン・ユニット78及びセンサがパイプライン74がいつ終了されるかを決定すべく利用できる数多くの従来技術がある。例えば、以下に議論されるように、シーケンス・マネージャ148は対応するSYNCパルス或は事象をセンサに提供し得る。代替的には、もしパイプライン74が実行する処理がタイムクリティカルでなければ、センサはパイプライン・バス50（図3）を介して事象をシーケンス・マネージャ148に送信し得る。

#### 【0084】

シーケンス・マネージャ148もホストプロセッサ42（図3）等のピアにSYNCパルス或は事象を生成することによってハードウェアに組み込まれたパイプライン74、74の動作に関する情報を提供し得る。シーケンス・マネージャ148はSYNCインターフェース150及び専用ライン（不図示）を介してSYNCパルスを送信し、出力事象キュー130及び出力データ・ハンドラー126を介して事象を送信する。先の例で参照されるように、ピアがパイプライン74からのデータ・ブロックを更に処理すると仮定する。シーケンス・マネージャ148は、SYNCパルス或は事象を介して、パイプライン74がデータから成るブロックをいつ処理し終えたかをそのピアに通知する。シーケンス・マネージャ148も、対応するSYNCパルス或は事象を生成して適切なピア（単数或は複数）に送信することによって、SYNCパルス或は事象の受領を確認し得る。

#### 【0085】

更に図5で参照されるようにパイプライン・ユニット78の動作は本発明の実施例に従って議論される。

#### 【0086】

データに対して、工業規格バス・インターフェース91はパイプライン・バス50（及び、もし有ればルータ61）からデータ信号（ホストプロセッサ42（図3）等のピアに起因する）を受信し、それら信号をヘッダー及びペイロードを各々が有するメッセージに変換する。

#### 【0087】

次に、工業規格バス・アダプタ118はその工業規格バス・インターフェース91からのメッセージを入力データ・ハンドラー120と互換性があるフォーマットに変換する。

#### 【0088】

次いで、入力データ・ハンドラー120はそのメッセージ・ヘッダを細かく調べて、各ヘッダーからデータ・ペイロードを記述する部分を抽出する。例えば、抽出されたヘッダ一部分は、例えば、パイプライン・ユニット78のアドレス、ペイロード中のデータのタイプ、或は、そのデータが意図されているパイプライン78、78を識別する例証識別子を含み得る。

#### 【0089】

次に、有効マネージャ 134 はその抽出ヘッダー部分を分析し、そのデータがハードウェアに組み込まれたパイプライン 74<sub>1</sub>~74<sub>n</sub> の内の 1 つに意図されていることを確認し、インターフェース 132 はそのデータをポート 106 を介して D P S R A M 100 の箇所に書き込み、そして、入力データ・ハンドラー 120 はその箇所と対応するデータ識別子に対するポインタを入力データ・キュー 122 に記憶する。データ識別子はデータが意図されているパイプライン或は複数のパイプライン 74<sub>1</sub>~74<sub>n</sub> を識別するか、或は、シーケンス・マネージャ 148 に以下に議論されるようにその識別を行わせることを可能とする情報を含む。代替的には、キュー 122 は各パイプライン 74<sub>1</sub>~74<sub>n</sub> に対する各サブキュー（不図示）を含み得て、入力データ・ハンドラー 120 は意図されたパイプライン或は意図された複数のパイプラインのサブキュー或は複数のサブキュー内にポインタを記憶する。この代替例において、データ識別子は省略し得る。更には、もしデータがメッセージのペイロードであれば、入力データ・ハンドラー 120 はインターフェース 132 が D P S R A M 100 内にそのデータを記憶する前にメッセージからそのデータを抽出する。代替的には、先に議論されたように、インターフェース 132 はメッセージ全体を D P S R A M 100 内に記憶し得る。

10

【0090】

次いで、適時に、シーケンス・マネージャ 148 は入力データ・キュー 122 からポインタ及びデータ識別子を読み取って、そのデータ識別子から、データが意図されているパイプライン或は複数のパイプライン 74<sub>1</sub>~74<sub>n</sub> を決定し、そのポインタを通信シェル 84 を介してそのパイプライン或は複数のパイプラインに渡す。

20

【0091】

次に、データ受信パイプライン或は複数のデータ受信パイプライン 74<sub>1</sub>~74<sub>n</sub> はインターフェース 140 にポート 108 を介して D P S R A M 100 の指された箇所からデータを検索させる。

【0092】

次いで、データ受信パイプライン或は複数のデータ受信パイプライン 74<sub>1</sub>~74<sub>n</sub> は検索されたデータを処理し、インターフェース 142 はその処理されたデータをポート 110 を介して D P S R A M 102 のある箇所に書き込み、そして、通信シェル 84 は出力データ・キュー 128 にその処理されたデータを指すポインタとその処理されたデータに対するデータ識別子とをロードする。そのデータ識別子は、その処理データに加入しているホストプロセッサ 42（図 3）等の仕向先ピア或は複数の仕向先ピアを識別するか、加入マネージャ 138 に引き続き仕向先ピア或は複数の仕向先ピア（例えば、図 3 のホストプロセッサ 42）を決定させることを可能とする情報（データ・タイプ等）を含む。代替的には、キュー 128 は各パイプライン 74<sub>1</sub>~74<sub>n</sub> に対する各サブキュー（不図示）を含み得て、通信シェル 84 は起因パイプライン或は複数の起因パイプラインのサブキュー或は複数のサブキューにポインタを記憶する。この代替例において、通信シェル 84 はデータ識別子のキュー 128 へのローディングを省略し得る。更には、もしパイプライン或は複数のパイプライン 74<sub>1</sub>~74<sub>n</sub> が検索されたデータを処理している間に中間データを生成すれば、インターフェース 144 はその中間データをポート 114 を介して D P S R A M 104 に書き込み、インターフェース 146 はポート 116 を介してその D P S R A M 104 から中間データを検索する。

30

40

【0093】

次に、出力データ・ハンドラー 126 は出力データ・キュー 128 からポインタ及びデータ識別子を検索し、加入マネージャ 138 はその識別子からデータの仕向先ピア或は複数の仕向先ピア（たとえば、図 3 のホストプロセッサ 42）を決定し、インターフェース 136 はポート 112 を介して D P S R A M 102 の指された箇所からデータを検索し、出力データ・ハンドラーはそのデータを工業規格バス・アダプタ 118 に送信する。もし仕向先ピアがメッセージのペイロードであるデータを必要とすれば、出力データ・ハンドラー 126 はそのメッセージを生成してそのメッセージをアダプタ 118 に送信する。例えば、データが多数の仕向先ピアを有して、パイプライン・バス 50 がメッセージ放送を

50

支援すると仮定する。出力データ・ハンドラー126はその仕向先ピア全てのアドレスを含む単一ヘッダーを生成し、そのヘッダー及びデータを1つのメッセージとして組み合わせて、単一メッセージを仕向先ピアの全てに（アダプタ118及び工業規格バス・インターフェース91を介して）同時に送信する。代替的には、出力データ・ハンドラー126は各ヘッダー、よって各メッセージを各仕向先ピアに対して生成し、それらメッセージの各々を個別に送信する。

#### 【0094】

次いで、工業規格バス・アダプタ118は出力データ・ハンドラー126からのデータをフォーマットして、工業規格バス・インターフェース91と互換性を持たせるように為す。

#### 【0095】

次に、工業規格バス・インターフェース91は工業規格バス・アダプタ118からのデータをフォーマットして、パイプライン・バス50（図3）と互換性を持たせるように為す。

#### 【0096】

随伴データを伴わない事象、即ちドアベルに対して、工業規格バス・インターフェース91はパイプライン・バス50から（もしあればルータ61からも）信号（図3のホストプロセッサ42等のピアに起因する）を受信し、その信号を事象を含むヘッダー（即ち、データ無しメッセージ）に変換する。

#### 【0097】

次に、工業規格バス・アダプタ118は工業規格バス・インターフェース91からのヘッダーを入力データ・ハンドラー120と互換性があるフォーマットに変換する。

#### 【0098】

次いで、入力データ・ハンドラー120はそのヘッダーから事象と事象の記述とを抽出する。例えば、記述は、例えば、パイプライン・ユニット78のアドレス、事象のタイプ、或は、その事象が意図されているパイプライン78<sub>1</sub>〜78<sub>n</sub>を識別する例証識別子を含み得る。

#### 【0099】

次に、有効マネージャ134は事象記述を分析し、その事象がハードウェアに組み込まれたパイプライン74<sub>1</sub>〜74<sub>n</sub>の内の1つに意図されていることを確認して、入力データ・ハンドラー120はその事象とその記述とを入力事象・キュー124に記憶する。

#### 【0100】

次いで、適時に、シーケンス・マネージャ148は入力事象キュー124からの事象及びその記述を読み取り、その事象に応じて、先に議論されたように、パイプライン74<sub>1</sub>〜74<sub>n</sub>の内の1つ或はそれ以上の動作をトリガーする。例えば、シーケンス・マネージャ148はパイプライン74<sub>2</sub>をトリガーし得て、パイプライン74<sub>1</sub>が先行してDPSRAM104に記憶したデータの処理を始める。

#### 【0101】

事象を出力すべく、シーケンス・マネージャ148は事象とその事象の記述を生成し、それら事象及び記述を出力事象キュー130にロードする。事象記述は、もし2つ以上の可能性ある仕向先ピアが存在する場合にその仕向先ピア（単数或は複数）を識別する。例えば、先に議論されたように、事象は入力事象、入力データ或は入力事象のメッセージ、或は、SYNCパルスの受領及び具現化を確認し得る。

#### 【0102】

次に、出力データ・ハンドラー126は事象とその記述を出力事象キュー130から検索し、加入マネージャ138はその事象記述から該事象の仕向先ピア或は複数の仕向先ピア（例えば、図3のホストプロセッサ42）を決定し、出力データ・ハンドラーは、先に議論されたように、その事象を工業規格バス・アダプタ118及び工業規格バス・インターフェース91を介して適切な仕向先ピア或は複数の適切な仕向先ピアに送信する。

#### 【0103】

コンフィギュレーション・コマンドに対して、工業規格バス・アダプタ 118 は工業規格バス・インターフェース 91 を介してホストプロセッサ 42 (図 3) からコマンドを受信し、そのコマンドをデータ無し事象 (即ち、ドアベル) に対して先に議論されたものと同様の方式で入力データ・ハンドラー 120 に提供する。

#### 【0104】

次に、有効マネージャ 134 はそのコマンドがパイプライン・ユニット 78 に対して意図されていることを確認し、入力データ・ハンドラー 120 はそのコマンドをコンフィギュレーション・マネージャ 90 にロードする。更に、入力データ・ハンドラー 120 或はコンフィギュレーション・マネージャ 90 の何れかもそのコマンドを出力データ・ハンドラー 126 に渡すことができ、それがパイプライン・ユニット 78 がコマンドを受信したことをそのコマンドを送信したピア (例えば、図 3 のホストプロセッサ 42) に送信し戻すことによって確認する。この確認技術はしばしば「エコー」と呼称される。

10

#### 【0105】

次いで、コンフィギュレーション・マネージャ 90 はそのコマンドを具現化する。例えば、コマンドはコンフィギュレーション・マネージャ 90 にデバッグ目的でパイプライン 74<sub>1</sub> - 74<sub>n</sub> の内の 1 つをディスエーブルさせ得る。或は、コマンドはホストプロセッサ 42 (図 3) 等のピアに出力データ・ハンドラー 126 を介してコンフィギュレーション・マネージャ 90 からパイプライン回路 80 の現行コンフィギュレーションを読み取らせることができる。加えて、コンフィギュレーション・コマンドを使用し得て、例外マネージャ 88 によって認識される例外を規定することができる。

20

#### 【0106】

例外に対して、パイプライン回路 80 の入力データ・キュー 122 等の構成要素は例外マネージャ 88 に向けて例外をトリガーする。一具現化例において、その構成要素は当該構成要素をモニタして、所定条件或は各種条件から成る組に応じてその例外をトリガーする例外トリガリング・アダプタ (不図示) を含む。例外トリガリング・アダプタは一度で設計され得てから例外を生成するパイプライン回路 80 の各構成要素の一部として含まれ得る普遍的回路であり得る。

#### 【0107】

次に、例外トリガーに応じて、例外マネージャ 88 は例外識別子を生成する。例えば、その識別子は入力データ・キュー 122 がオーバーフローしたことを示し得る。更には、その識別子は、もし 2 つ以上の可能性ある仕向先ピアが存在する場合にその仕向先ピアを含み得る。

30

#### 【0108】

次いで、出力データ・ハンドラー 126 は例外マネージャ 88 から例外識別子を検索し、その例外識別子を、先行して引用された「改訂された計算アーキテクチャを有する計算マシン、関連システム、並びに、方法」と題された特許文献 3 で議論されたようにホストプロセッサ 42 (図 3) に送信する。代替的には、もし多数の可能性ある仕向先ピアが存在すれば、例外識別子も仕向先情報を含むことができ、それから加入マネージャ 138 がその識別子の仕向先ピア或は複数の仕向先ピア (例えば、図 3 のホストプロセッサ 42) を決定する。出力データ・ハンドラー 126 は、次いで、その識別子を、工業規格バス・アダプタ 118 及び工業規格バス・インターフェース 91 を介して、仕向先ピア或は複数の仕向先ピアに送信する。

40

#### 【0109】

更に図 5 で参照されるように、パイプライン・ユニット 78 に対する代替実施例が存在する。例えば、DP SRAM を含むものとして説明されているが、データ・メモリ 92 はクワッド-データ-レート (QDR) SRAM 等のメモリ IC の他のタイプを含み得る。

#### 【0110】

図 6 は、本発明の実施例に従った図 5 のインターフェース 142 のブロック線図である。図 5 と連携して先に議論されたように、インターフェース 142 はハードウェアに組み込まれたパイプライン 74<sub>1</sub> - 74<sub>n</sub> から DP SRAM 102 に処理データを書き込む。以

50

下に議論されるように、インターフェース142の構造は、データ「ボトルネック」を低減又はなくし、パイプライン回路80（図5）がPLICである場合、PLICのローカル及びグローバルのルーティング・リソースを効率的に利用する。

#### 【0111】

インターフェース142は書き込みチャンネル150<sub>1</sub>〜150<sub>n</sub>を含み、各ハードウェアに組み込まれたパイプライン74<sub>1</sub>〜74<sub>n</sub>（図5）に対して1つずつのチャンネルとなっており、コントローラ152を含む。図示の簡略化の目的のため、チャンネル150<sub>1</sub>が以下に議論され、他のチャンネル150<sub>2</sub>〜150<sub>n</sub>の動作及び構造が別段の説明がない限り同様であることを理解して頂きたい。

#### 【0112】

チャンネル150<sub>1</sub>は書き込みアドレス／データFIFO154<sub>1</sub>及びアドレス／データレジスタ156<sub>1</sub>を含む。

#### 【0113】

FIFO154<sub>1</sub>は、パイプライン74<sub>1</sub>がDPSRAM102に書き込むデータを記憶し、コントローラ152がレジスタ156<sub>1</sub>を介してDPSRAM102に実際にデータを書き込むことができるまで、パイプラインがデータを書き込むDPSRAM102内の箇所のアドレスを記憶する。それ故に、FIFO154<sub>1</sub>は、もしコントローラ152が先行するデータの書き込みを終了するまでに、パイプライン74<sub>1</sub>がチャンネル150<sub>1</sub>へのデータ書き込みを「待機」しなければならなかった場合に生じ得るデータ・ボトルネックを低減又はなくする。

#### 【0114】

FIFO154<sub>1</sub>はバス158<sub>1</sub>を介してパイプライン74<sub>1</sub>からデータを受信し、バス160<sub>1</sub>を介してデータが書き込まれることになる箇所のアドレスを受信し、そのデータ及びアドレスをバス162<sub>1</sub>及び164<sub>1</sub>を介してレジスタ156<sub>1</sub>に提供する。更には、FIFO154<sub>1</sub>はパイプライン74<sub>1</sub>からライン166<sub>1</sub>上の書き込みFIFO信号（WRITE FIFO信号）を受信し、ライン168<sub>1</sub>を介してクロック信号（CLOCK信号）を受信し、そしてFIFO満満信号をライン170<sub>1</sub>上でパイプライン74<sub>1</sub>に提供する。加えて、FIFO154<sub>1</sub>はライン172<sub>1</sub>を介してコントローラ152から読み取りFIFO信号を受信して、ライン174<sub>1</sub>を介してFIFO EMPTY信号（FIFO空信号）をコントローラに提供する。パイプライン回路80（図5）がPLICである場合、バス158<sub>1</sub>、160<sub>1</sub>、162<sub>1</sub>、164<sub>1</sub>及びライン166<sub>1</sub>、168<sub>1</sub>、170<sub>1</sub>、172<sub>1</sub>、174<sub>1</sub>は、好ましくはローカル・ルーティング・リソースを用いて形成される。典型的には、ローカル・ルーティング・リソースは、信号経路長が一般により短く且つルーティングの具現化がより容易であるため、グローバル・ルーティング・リソースよりも好まれる。

#### 【0115】

レジスタ156<sub>1</sub>は、バス162<sub>1</sub>及び164<sub>1</sub>を介して、FIFO154<sub>1</sub>から書き込まれるべきデータと、書き込み箇所のアドレスとをそれぞれ受信し、それらデータ及びアドレスをアドレス／データバス176<sub>1</sub>を介してDPSRAM102（図5）のポート110に提供する。更には、レジスタ156<sub>1</sub>もデータ及びアドレスを、以下に議論されるように、アドレス／データバス178<sub>1</sub>を介してレジスタ156<sub>2</sub>〜156<sub>n</sub>から受信する。加えて、レジスタ156<sub>1</sub>はライン180<sub>1</sub>を介してコントローラ152からシフト／ロード信号（SHIFT/LOAD信号）を受信する。パイプライン回路80（図5）がPLICである場合、バス176<sub>1</sub>は典型的にはグローバル・ルーティング・リソースを用いて形成され、バス178<sub>1</sub>〜178<sub>n</sub>及びライン180<sub>1</sub>は好ましくはローカル・ルーティング・リソースを用いて形成される。

#### 【0116】

FIFO空信号の受信と読み取りFIFO及びシフト／ロード信号の生成に加えて、コントローラ152は書き込みDPSRAM信号（WRITE DPSRAM信号）をライン182<sub>1</sub>を介してDPSRAM102（図5）のポート110に提供する。



## 【0117】

更に図6で参照されるように、インターフェース142の動作が議論される。

## 【0118】

先ず、FIFO154<sub>1</sub>はFIFOの現行状態（「充滿」或は「非充滿」）と対応する論理レベルまでFIFO充滿信号を駆動する。

## 【0119】

次に、もしFIFO154<sub>1</sub>が充滿していなく且つパイプライン74<sub>1</sub>が書き込むべきデータを処理していれば、パイプラインはデータ及び対応するアドレスをバス158<sub>1</sub>及び160<sub>1</sub>までそれぞれ駆動し、書き込み信号をアサートし、よってデータ及びアドレスをFIFOにロードする。しかしながら、もしFIFO154<sub>1</sub>が充滿していれば、パイプライン74<sub>1</sub>はデータをロードする前にFIFOが非充滿となるまで待機する。 10

## 【0120】

次いで、FIFO154<sub>1</sub>はFIFO空信号をFIFOの現行状態（「空」或は「非空」）と対応する論理レベルまで駆動する。

## 【0121】

次に、もしFIFO154<sub>1</sub>が非空であれば、コントローラ152は読み取りFIFO信号をアサートし、シフト/ロード信号をロード論理レベルまで駆動し、よってFIFOから最初にロードされたデータ及びアドレスをレジスタ156<sub>1</sub>にロードする。もしFIFO154<sub>1</sub>が空であれば、コントローラ152は読み取りFIFOをアサートしないが、他のFIFO154<sub>2</sub>〜154<sub>n</sub>の内の任意のものが空でなければ、シフト・ロードをロード論理レベルまで駆動する。 20

## 【0122】

チャネル150<sub>2</sub>〜150<sub>n</sub>は同様の方式で動作して、FIFO154<sub>2</sub>〜154<sub>n</sub>に最初にロードされたデータがそれぞれレジスタ156<sub>2</sub>〜156<sub>n</sub>にロードされるように為す。

## 【0123】

次いで、コントローラ152はシフト/ロード信号をシフト論理レベルまで駆動し、書き込みDPSRAM信号をアサートし、よってレジスタ156<sub>1</sub>〜156<sub>n</sub>からアドレス/データ・バス176にデータ及びアドレスを順次シフトし、DPSRAM102の対応する箇所にデータをロードする。詳細には、第1シフト・サイクル中、レジスタ156<sub>1</sub>からのデータ及びアドレスはバス176にシフトされて、FIFO154<sub>1</sub>からのデータはDPSRAM102のアドレス指定された箇所にロードされるように為す。また第1シフト・サイクル中、レジスタ156<sub>2</sub>からのデータ及びアドレスはレジスタ156<sub>1</sub>にシフトされて、レジスタ156<sub>2</sub>（不図示）からのデータ及びアドレスはレジスタ156<sub>1</sub>にシフトされる、等々である。第2シフト・サイクル中、レジスタ156<sub>1</sub>からのデータ及びアドレスはバス176にシフトされて、FIFO154<sub>2</sub>からのデータはDPSRAM102のアドレス指定された箇所にロードされるように為す。また第2シフト・サイクル中、レジスタ156<sub>2</sub>からのデータ及びアドレスはレジスタ156<sub>1</sub>にシフトされ、レジスタ156<sub>3</sub>（不図示）からのデータ及びアドレスはレジスタ156<sub>2</sub>にシフトされる、等々である。n個のシフト・サイクルがあって、第n番目のシフト・サイクル中、レジスタ156<sub>n</sub>からのデータ及びアドレス（FIFO154<sub>n</sub>からのデータ及びアドレスである）はバス176にシフトされる。コントローラ152はシフト/ロード信号を脈動することによって、或は、レジスタ156<sub>1</sub>〜156<sub>n</sub>と結合されているシフト・クロック信号（不図示）を生成することによってそれらシフト・サイクルを具現化し得る。更には、対応するFIFO154<sub>1</sub>〜154<sub>n</sub>がコントローラ152がレジスタにロードした際に空であるので、レジスタ156<sub>1</sub>〜156<sub>n</sub>の内の1つが特定のシフト動作中に空であれば、コントローラはその空レジスタを迂回し得て、よって、マルチデータ及びマルチアドレスのバス176へのシフトを回避することによってシフト動作を短縮する。 30 40

## 【0124】

図5及び図6で参照されるように、本発明の実施例に従えば、インターフェース144はインターフェース142と類似し、インターフェース132も該インターフェース13 50

2が1つの書き込みチャネル150のみを含むことを除いてインターフェース142と類似している。

#### 【0125】

図7は、本発明の実施例に従った図5のインターフェース140のブロック線図である。図5と連携して先に議論されたように、インターフェース140はDPSRAM100から入力データを読み取って、そのデータをハードウェアに組み込まれた74<sub>1</sub>〜74<sub>4</sub>に転送する。以下に議論されるように、インターフェース140の構造はデータ「ボトルネック」を低減或はなくし、パイプライン回路80（図5）がPLICである場合、PLICのローカル及びグローバルのルーティング・リソースを効率的に利用する。

#### 【0126】

インターフェース140は、各ハードウェアに組み込まれたパイプライン74<sub>1</sub>〜74<sub>4</sub>（図5）に対して1つずつのチャネルである読み取りチャネル190<sub>1</sub>〜190<sub>4</sub>とコントローラ192とを含む。図示の簡略化のため、読み取りチャネル190<sub>1</sub>が以下に議論されているが、他の読み取りチャネル190<sub>2</sub>〜190<sub>4</sub>の動作及び構造は別段の説明がない限り類似していることを理解していただきたい。

#### 【0127】

チャネル190<sub>1</sub>はFIFO194<sub>1</sub>及びアドレス／識別子（ID）レジスタ196<sub>1</sub>を含む。以下に議論されるように、識別子は、データを受信すべくDPSRAM100の特定箇所からそのデータを読み取る要求を為すパイプライン74<sub>1</sub>〜74<sub>4</sub>を識別する。

#### 【0128】

FIFO194<sub>1</sub>は2つのサブFIFO（不図示）を含み、一方がパイプライン74<sub>1</sub>が入力データを読み取ることを望んでいるDPSRAM100内の箇所のアドレスを記憶し、他方がDPSRAM100から読み取られたデータを記憶する。それ故に、FIFO194<sub>1</sub>は、もしパイプライン74<sub>1</sub>が、コントローラ192が先行するデータの読み取りを終了するまで、チャネル190<sub>1</sub>に読み取りアドレスを提供するために「待機」しなければならない場合、或は、コントローラが、コントローラが引き続くデータを読み取る前にパイプライン74<sub>1</sub>が読み取りデータを検索するまで待機しなければならない場合に生じ得るボトルネックを低減或はなくする。

#### 【0129】

FIFO194<sub>1</sub>はバス198<sub>1</sub>を介してパイプライン74<sub>1</sub>から読み取りアドレスを受信し、そのアドレス及びIDをバス200<sub>1</sub>を介してレジスタ196<sub>1</sub>に提供する。IDはパイプライン74<sub>1</sub>と対応して典型的には変化しないので、FIFO194<sub>1</sub>はそのIDを記憶し得て、そのIDをアドレスとつなぎ合わせる。代替的には、パイプライン74<sub>1</sub>はIDをバス198<sub>1</sub>を介してFIFO194<sub>1</sub>に提供し得る。更には、FIFO194<sub>1</sub>はライン202<sub>1</sub>を介してパイプライン74<sub>1</sub>から準備書き込みFIFO信号を受信し、ライン204<sub>1</sub>を介してCLOCK信号を受信し、（読み取りアドレスの）FIFO充満信号をライン206<sub>1</sub>を介してパイプラインに提供する。加えて、FIFO194<sub>1</sub>はライン208<sub>1</sub>を介してコントローラ192から書き込み／読み取りFIFO信号（WRITE／READ FIFOSIGNAL）を受信し、FIFO空信号をライン210<sub>1</sub>を介してコントローラに提供する。更には、FIFO194<sub>1</sub>はバス212<sub>1</sub>を介してコントローラ192から読み取りデータ及び対応するIDを受信し、そのデータをバス214<sub>1</sub>を介してパイプライン74<sub>1</sub>に提供する。パイプライン回路80（図5）がPLICである場合、バス198<sub>1</sub>、200<sub>1</sub>、214<sub>1</sub>及びライン202<sub>1</sub>、204<sub>1</sub>、206<sub>1</sub>、208<sub>1</sub>、210<sub>1</sub>は好ましくはローカル・ルーティング・リソースを用いて形成され、バス212<sub>1</sub>は典型的にはグローバル・ルーティング・リソースを用いて形成される。

#### 【0130】

レジスタ196<sub>1</sub>は読み取られるべき箇所のアドレスと対応するIDをバス206<sub>1</sub>を介してFIFO194<sub>1</sub>から受信し、そのアドレスをアドレス・バス216<sub>1</sub>を介してDPSRAM100（図5）のポート108に提供して、そのIDをバス218<sub>1</sub>を介してコントローラ192に提供する。更には、レジスタ196<sub>1</sub>も、以下に議論されるように、アド



レス／IDバス220<sub>1</sub>を介してレジスタ196<sub>2</sub>～196<sub>3</sub>からアドレス及びIDを受信する。加えて、レジスタ196<sub>1</sub>はライン222を介してコントローラ192からシフト／ロード信号を受信する。パイプライン回路80（図5）がPLICである場合、バス216は典型的にはグローバル・ルーティング・リソースを用いて形成され、バス220<sub>1</sub>～220<sub>3</sub>、及びライン222は好ましくはローカル・ルーティング・リソースを用いて形成される。

#### 【0131】

FIFO空信号の受信、書き込み／読み取りFIFO及びシフト／ロード信号の生成、並びに、読み取りデータ及び対応するIDの提供に加えて、コントローラ192はバス224を介してDPSSRAM100（図5）のポート108から読み取られたデータを受信し、読み取りDPSSRAM信号（READ DPSSRAM信号）をライン226上に生成するが、そのラインがこの信号をポート108と結合する。パイプライン回路80（図5）がPLICの場合、バス224及びライン226は典型的にはグローバル・ルーティング・リソースを用いて形成される。

#### 【0132】

更に図7で参照されるように、インターフェース140の動作が議論されている。

#### 【0133】

先ず、FIFO194<sub>1</sub>は、読み取りアドレスに対するFIFOの現行状態（「充填」或は「非充填」）と対応する論理レベルまでFIFO充填信号を駆動する。即ち、もしFIFO194<sub>1</sub>が読み取られるべきアドレスで充填していれば、FIFO充填の論理レベルを1つのレベルまで駆動し、もしFIFOが読み取りアドレスで充填していなければ、FIFO充填の論理レベルを別のレベルまで駆動する。

#### 【0134】

次に、もしFIFO194<sub>1</sub>が読み取りアドレスで充填していなく且つパイプライン74<sub>1</sub>がより多くの入力データを処理すべく準備がされると、パイプラインは読み取られるべきデータのアドレスをバス198<sub>1</sub>まで駆動し、読み取り／書き込みFIFO信号（READ／WRITE FIFO信号）を書き込みレベルまでアサートし、よってアドレスをFIFOにロードする。図5と連携して先に議論されたように、パイプライン74<sub>1</sub>はシーケンス・マネージャ148を介して入力データ・キュー122からアドレスを得る。しかしながらFIFO194<sub>1</sub>が読み取りアドレスで充填していれば、パイプライン74<sub>1</sub>は読み取りアドレスのローディング前にFIFOが充填しなくなるまで待機する。

#### 【0135】

次いで、FIFO194<sub>1</sub>はFIFO空信号を読み取りアドレスに対するFIFOの現行状態（「空」或は「非空」）と対応する論理レベルまで駆動する。即ち、もしFIFO194<sub>1</sub>が少なくとも1つの読み取りアドレスでロードされれば、FIFO空の論理レベルを1つのレベルまで駆動し、もしFIFOが読み取りアドレス無しでロードされれば、FIFO空の論理レベルを別のレベルまで駆動する。

#### 【0136】

次に、もしFIFO194<sub>1</sub>が空でなければ、コントローラ192は書き込み／読み取りFIFO信号を読み取り論理レベルまでアサートし、シフト／ロード信号をロード論理レベルまで駆動し、よってFIFOから最初にロードされたアドレスとIDをレジスタ196<sub>1</sub>にロードする。

#### 【0137】

チャネル190<sub>2</sub>～190<sub>3</sub>は同様の方式で動作して、コントローラ192がFIFO194<sub>2</sub>～194<sub>3</sub>から最初にロードされたアドレスとIDをレジスタ196<sub>2</sub>～196<sub>3</sub>にそれぞれロードする。もしFIFO194<sub>2</sub>～194<sub>3</sub>の全てが空であれば、コントローラ192は前進の前にアドレスを受信すべくFIFOの内の少なくとも1つを待機する。

#### 【0138】

次いで、コントローラ192はシフト／ロード信号をシフト論理レベルまで駆動し、読み取りDPSSRAM信号をアサートしてレジスタ196<sub>1</sub>～196<sub>3</sub>からアドレス及びID

バス216及び218にアドレス及びIDを順次シフトして、バス224を介してDPSRAM100の対応する箇所からデータを順次読み取る。

#### 【0139】

次に、コントローラ192は受信されたデータと対応するID（そのIDはFIFO194<sub>1</sub>、-194<sub>n</sub>の各々にそれがデータの意図された受取人であるか否かを決定させる）をバス212まで駆動し、書き込み／読み取りFIFO信号を書き込みレベルまで駆動し、よってそのデータを各FIFO194<sub>1</sub>、-194<sub>n</sub>に順次書き込む。

#### 【0140】

次いで、ハードウェアに組み込まれたパイプライン74<sub>1</sub>、-74<sub>n</sub>はそれらの読み取り／書き込みFIFO信号を読み取りレベルまで順次アサートし、そのデータをバス214<sub>1</sub>、-214<sub>n</sub>を介して順次読み取る。

#### 【0141】

更に図7で参照されるように、それらのデータ読み取り演算子のより詳細な議論が提示される。

#### 【0142】

第1シフト・サイクル中、コントローラ192はレジスタ196<sub>1</sub>からバス216及び218までアドレス及びIDをそれぞれシフトし、読み取りDPSRAMをアサートし、そうして、バス224を介してDPSRAM100の対応する箇所からデータを読み取り、バス218からIDを読み取る。次に、コントローラ192は書き込み／読み取りFIFO信号をライン208<sub>1</sub>上で書き込みレベルまで駆動し、受信されたデータとIDをバス212に駆動する。IDがFIFO194<sub>1</sub>からのIDであるので、FIFO194<sub>1</sub>はそのIDを認識し、よって書き込み／読み取りFIFO信号の書き込みレベルに応じてバス212からデータをロードする。残りのFIFO194<sub>2</sub>、-194<sub>n</sub>は、バス212上のIDがそれらのIDと対応しないので、データをロードしない。次いで、パイプライン74<sub>1</sub>は読み取り／書き込みFIFO信号をライン202<sub>1</sub>上で読み取りレベルまでアサートし、バス214<sub>1</sub>を介して読み取りデータを検索する。また、第1シフト・サイクル中、レジスタ196<sub>2</sub>からのアドレス及びIDはレジスタ196<sub>1</sub>にシフトされ、レジスタ196<sub>3</sub>（不図示）からのアドレス及びIDはレジスタ196<sub>2</sub>にシフトされ、等々である。代特的には、コントローラ192はIDを認識し、書き込み／読み取りFIFO信号だけをライン208<sub>1</sub>上で書き込みレベルまで駆動する。これはIDをFIFO194<sub>1</sub>、-194<sub>n</sub>まで送信すべきコントローラ192に対する必要性をなくする。別の代特例において、書き込み／読み取りFIFO信号は読み取り信号のみであり、FIFO194<sub>1</sub>（他のFIFO194<sub>2</sub>、-194<sub>n</sub>と共に）はバス212上のIDがFIFO194<sub>1</sub>のIDと符合する場合にバス212上にデータをロードする。これは書き込み信号を生成すべきコントローラ192の必要性をなくする。

#### 【0143】

第2シフト・サイクル中、レジスタ196<sub>1</sub>からのアドレス及びIDはバス216及び218にシフトされて、コントローラ192がFIFO194<sub>2</sub>によって特定されるDPSRAM100の箇所からデータを読み取るように為す。次に、コントローラ192は書き込み／読み取りFIFO信号を書き込みレベルまで駆動し、受信されたデータとIDをバス212に駆動する。IDがFIFO194<sub>2</sub>からのIDであるので、FIFO194<sub>2</sub>はそのIDを認識し、そうしてバス212からデータをロードする。残りのFIFO194<sub>1</sub>及び194<sub>3</sub>、-194<sub>n</sub>は、バス212上のIDがそれらのIDと対応しないのでデータをロードしない。次いで、パイプライン74<sub>2</sub>はその読み取り／書き込みFIFO信号を読み取りレベルまでアサートし、バス214<sub>2</sub>を介して読み取りデータを検索する。また第2シフト・サイクル中、レジスタ196<sub>2</sub>からのアドレス及びIDはレジスタ196<sub>1</sub>にシフトされ、レジスタ196<sub>3</sub>（不図示）からのアドレス及びIDはレジスタ196<sub>2</sub>にシフトされ、等々である。

#### 【0144】

これはn個のシフト・サイクルの間続行、即ち、レジスタ196<sub>n</sub>（FIFO194<sub>n</sub>が

らのアドレス及びID)からのアドレス及びIDがバス216及び218にそれぞれシフトされるまで続行する。コントローラ192はシフト/ロード信号を脈動することによって、或はレジスタ196<sub>1</sub>~196<sub>n</sub>と結合されたシフト・クロック信号(不図示)を生成することによってそれらシフト・サイクルを具現化し得る。更には、もし特定シフト・サイクル中にレジスタ196<sub>1</sub>~194<sub>n</sub>の内の1つが、その対応するFIFO194<sub>1</sub>~194<sub>n</sub>が空であるため空であれば、コントローラ192は空レジスタを迂回し、よってメル・アドレスのバス216へのシフトを回避することによってシフト動作を短縮する。

【0145】

図5及び図6で参照されるように、本発明の実施例に従えば、インターフェース144はインターフェース140と類似し、インターフェース136も該インターフェース136が1つのみの読み取りチャネル190を含むことを除けばインターフェース140と類似し、よってID回路を全く含まない。

【0146】

図8は、本発明の別の実施例に従った図4のパイプライン・ユニット230の概略ブロック図である。このパイプライン・ユニット230は、該パイプライン・ユニット230が多数のパイプライン回路80を含むことを除いて(ここでは2つのパイプライン回路80a及び80b)図4のパイプライン・ユニット78と類似している。パイプライン回路80の数を増大することは、典型的には、ハードウェアに組み込まれたパイプライン74<sub>1</sub>~74<sub>n</sub>の数nに関する増大を可能とし、よってパイプライン・ユニット78と比較してパイプライン・ユニット230の機能に関する増大を可能とする。

【0147】

図8のパイプライン・ユニット230において、サービス構成要素、即ち、通信インターフェース82、コントローラ86、例外マネージャ88、コンフィギュレーション・マネージャ90、並びに、任意選択的な工業規格バス・インターフェース91はパイプライン回路80a上に配置され、パイプライン74<sub>1</sub>~74<sub>n</sub>及び通信シェル84はパイプライン回路80b上に配置される。サービス構成要素及びパイプライン74<sub>1</sub>~74<sub>n</sub>を別々のパイプライン回路上に位置決めすることによって、サービス構成要素及びパイプラインが同一のパイプライン回路上に位置決めされている場合に可能であるものよりも、より多くの数nのパイプライン及び/或はより複雑なパイプラインを含み得る。代替的には、パイプライン74<sub>1</sub>~74<sub>n</sub>をインターフェース82にインターフェースする通信シェル84の部分とコントローラ86はパイプライン回路80a上に配置される。

【0148】

図9は、本発明の実施例に従った図8のパイプライン・ユニット230のパイプライン回路80a、80b及びデータ・メモリ92の概略ブロック図である。図9のパイプライン構成要素が2つのパイプライン回路上に配置されていること以外、パイプライン回路80a及び80bとメモリ92の構造及び動作は、図5のパイプライン回路80及びメモリ92のものと同じである。

【0149】

先行する議論は当業者が本発明を作製し使用することを可能とすべく提示されている。種々実施例への様々な変更は当業者には容易に明かであろうし、ここでの包括的な原則は本発明の精神及び範囲から逸脱することなしに他の実施例及び適用例に適用され得る。よって、本発明は図示された実施例に限定されることが意図されておらず、ここに開示された原理及び特徴と一貫した最も広い範囲と一致されるべきものである。

【図面の簡単な説明】

【0150】

【図1】図1は、従来の多数プロセッサ・アーキテクチャを有する計算マシンのブロック図である。

【図2】図2は、従来のハードウェアに組み込まれたパイプラインのブロック図である。

【図3】図3は、本発明の実施例に従ったビクタール・アーキテクチャを有する計算

マシンのブロック線図である。

【図4】図4は、本発明の実施例に従った図3のパイプライン加速器のブロック線図である。

【図5】図5は、本発明の実施例に従った図4のハードウェアに組み込まれたパイプライン回路とデータ・メモリとのブロック線図である。

【図6】図6は、本発明の実施例に従った図5の通信シェルのメモリ書き込みインターフェースのブロック線図である。

【図7】図7は、本発明の実施例に従った図5の通信シェルのメモリ読み取りインターフェースのブロック線図である。

【図8】図8は、本発明の別の実施例に従った図3のパイプライン加速器のブロック線図である。

10

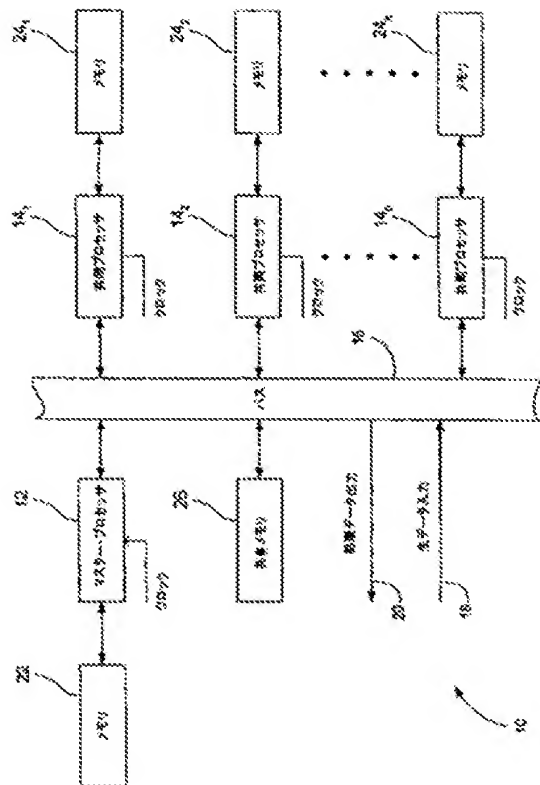
【図9】図9は、本発明の実施例に従った図8のハードウェアに組み込まれたパイプライン回路とデータ・メモリとのブロック線図である。

【符号の説明】

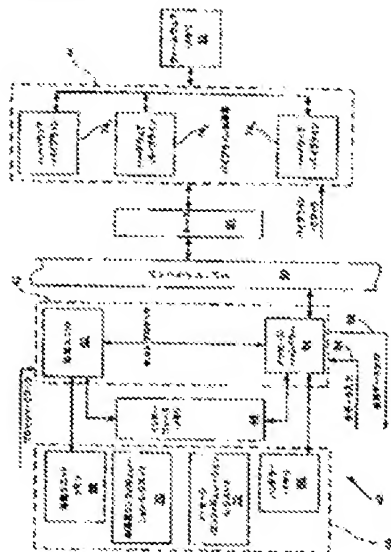
【0151】

10	計算マシン	
14	共同プロセッサ	
40	ピア-ベクトル・マシン	
42	ホストプロセッサ	
44	パイプライン加速器	20
46	プロセッサ・メモリ	
48	インターフェース・メモリ	
50	パイプライン・バス	
52	ファームウェア・メモリ	
54	生データ入力ポート	
58	処理データ出力ポート	
61	ルータ	
62	処理ユニット	
64	メッセージ・ハンドラー	
66	処理ユニット・メモリ	30
68	ハンドラー・メモリ	
70	加速器コンフィギュレーション・レジストリ	
72	メッセージ・コンフィギュレーション・レジストリ	
74	ハードウェアに組み込まれたパイプライン	
78	パイプライン・ユニット	
80	パイプライン回路	
86	パイプライン・コントローラ	
88	例外マネージャ	
90	コンフィギュレーション・マネージャ	
91	工業規格バス・インターフェース	40

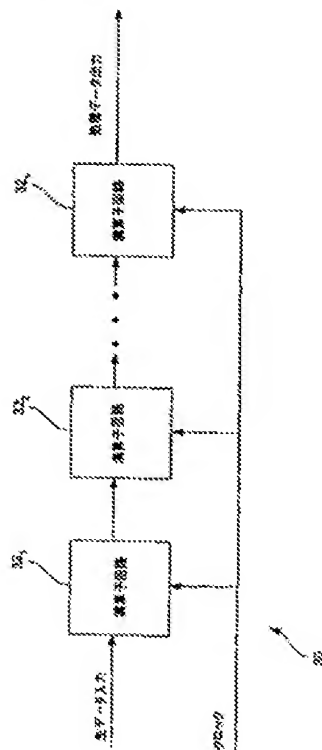
【図 1】



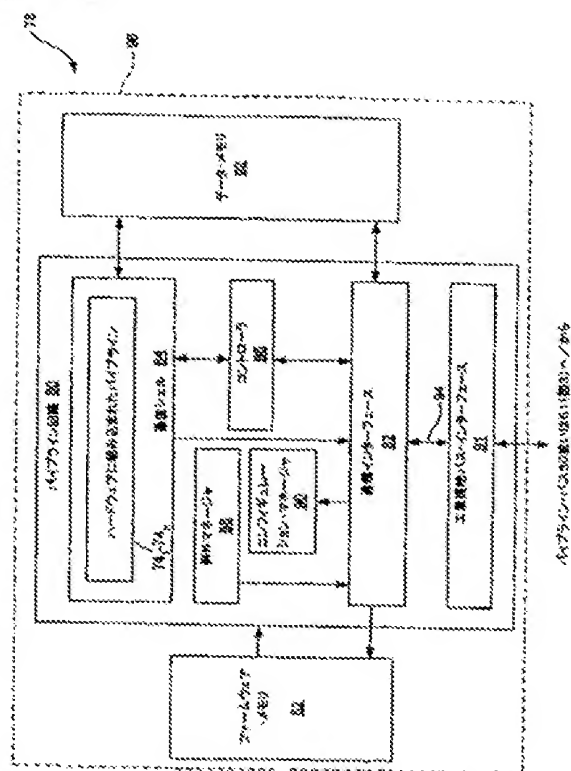
【図 3】



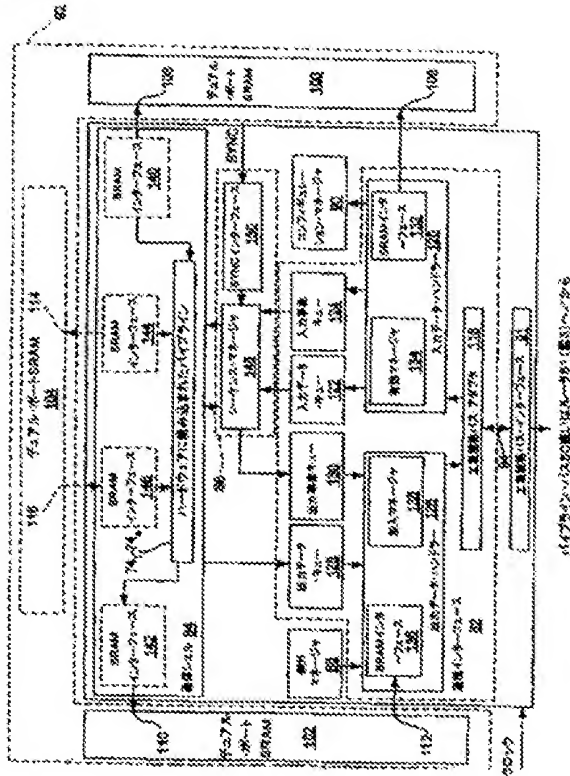
【図 2】



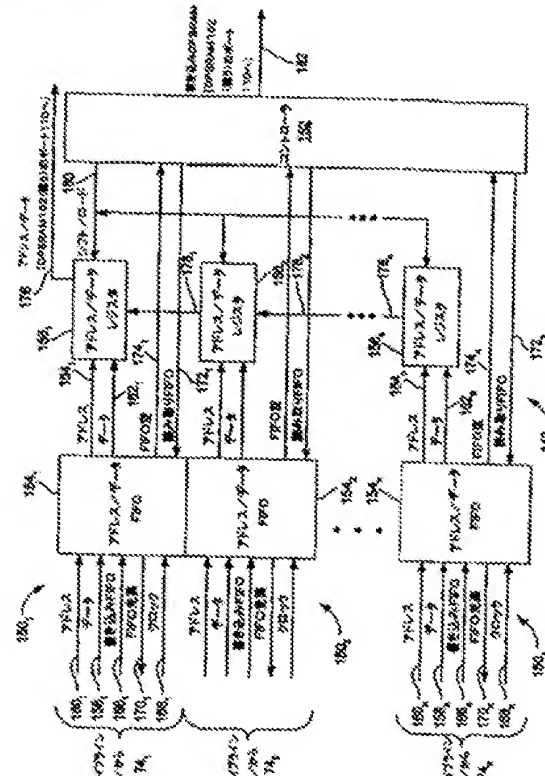
【図 4】



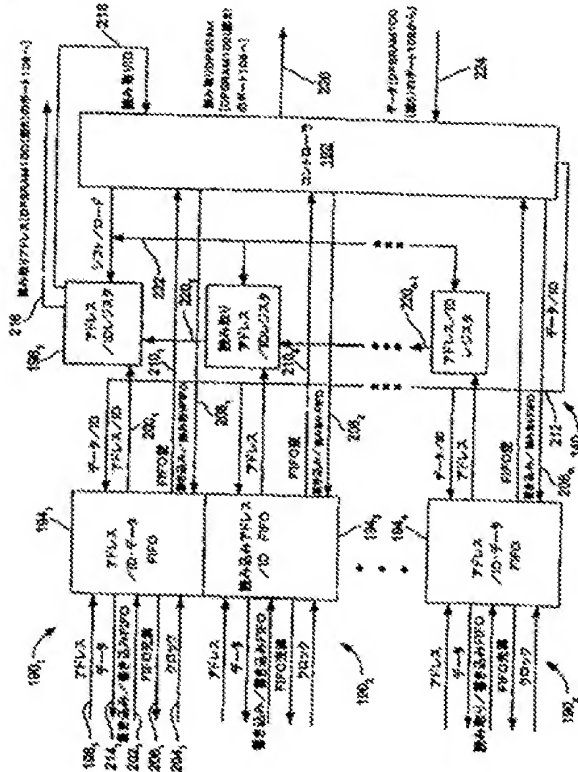
【図 5】



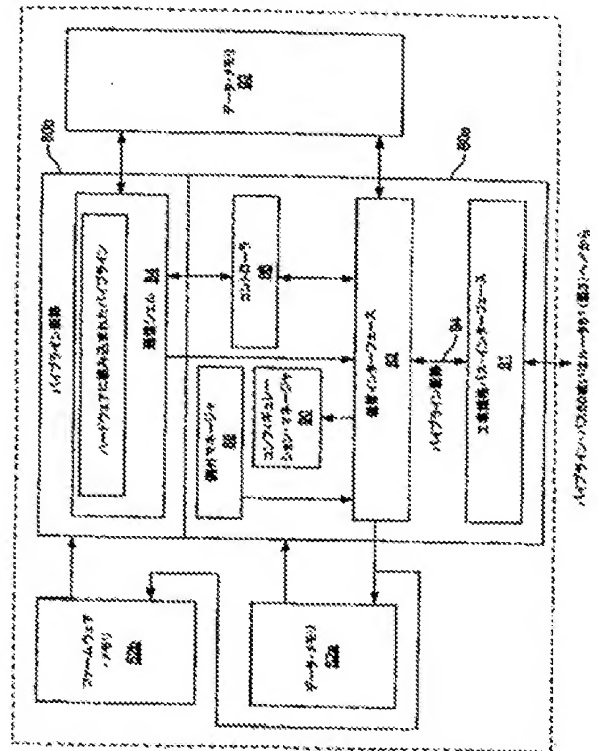
【図 6】



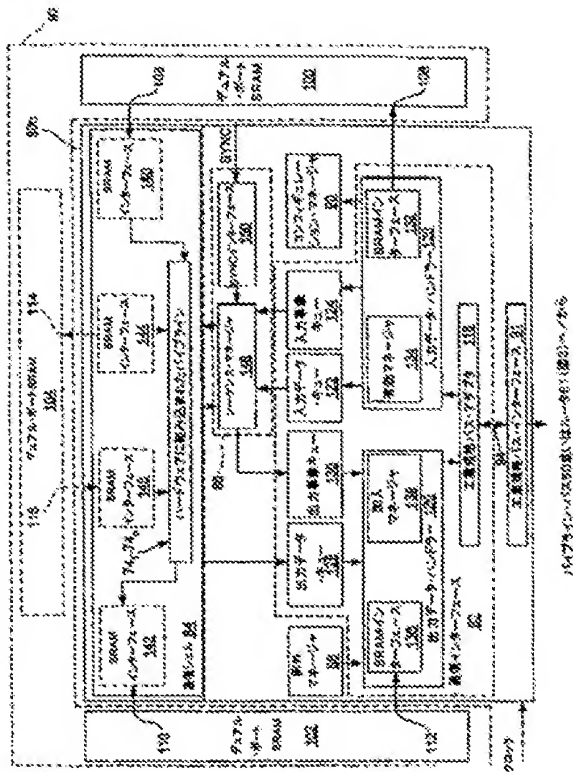
【図 7】



【図 8】



【附 9】



## 【国際調査報告】

## INTERNATIONAL SEARCH REPORT

PCT/US 03/34558

A. CLASSIFICATION OF SUBJECT MATTER		
IPC 7 G06F9/46 G06F9/38 G06F15/78		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols)		
IPC 7 G06F		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practical, search terms used)		
EPD-Internal		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 1 061 439 A (HEWLETT PACKARD CO) 20 December 2000 (2000-12-20) page 4 paragraph [0018] - paragraph [0021] paragraph [0028] - paragraph [0031] paragraph [0054] - paragraph [0056] -----	1-14, 41-50
X	US 5 892 962 A (CLOUTIER ET AL) 6 April 1999 (1999-04-06)  claim 7 figures 1,2 column 2, line 53 - line 61 column 3, line 34 - column 4, line 6 column 5, line 24 - line 29 ----- -/-	1-8, 41-43, 49,50
<input checked="" type="checkbox"/> Further documents are listed in the continuation of box C. <input checked="" type="checkbox"/> Patent family members are listed in annex.		
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier document not published on or after the international filing date "L" document which may throw doubt on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document relating to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "A" document member of the same patent family		
Date of the actual completion of the international search		Date of mailing of the international search report
18 February 2005		16 05 2005
Name and mailing address of the ISA European Patent Office, P.O. Box 1, Patentstrasse 1 69, 2000 NY (Frankfurt) Tel: (+31-70) 340-0340, Tx: 31 851 apr nl Fax: (+31-70) 340-0316		Authorized officer  Bosch Vivancos, P

Form PCT/ISA/210 (second sheet) January 2004

page 1 of 2



## INTERNATIONAL SEARCH REPORT

PCT/US 03/34558

C. (Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevance to claim No.
A	<p>SALCIC Z ET AL: "FLIX environment for generation of custom-configurable machines in FPLDs for embedded applications" MICROPROCESSORS AND MICROSYSTEMS, IPC BUSINESS PRESS LTD, LONDON, GB, vol. 23, no. 8-9, 15 December 1999 (1999-12-15), pages 513-526, XP004254077  ISSN: 0141-9331  page 520, right-hand column, line 1 - page 521, left-hand column, line 10  -----</p>	<p>1-14,  41-50</p>

Form PCT/US/8110 (publication of second sheet) (January 2004)

page 2 of 2

## INTERNATIONAL SEARCH REPORT

PCT/US 03/34558  
16

## Box I Observations where certain claims were found unsearchable (Continuation of Item 1 of first sheet)

This International Search Report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:  
because they relate to subject matter not required to be searched by this Authority, namely:
2. ☐ Claims Nos.:  
because they relate to parts of the International Application that do not comply with the prescribed requirements to such an extent that no meaningful International Search can be carried out, specifically:
3. ☐ Claims Nos.:  
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 8.4(a).

## Box II Observations where unity of invention is lacking (Continuation of Item 2 of first sheet)

This International Searching Authority found multiple inventions in this International application, as follows:

see additional sheet

1. ☐ As all required additional search fees were timely paid by the applicant, this International Search Report covers all searchable claims.
2. ☐ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.
3. ☐ As only some of the required additional search fees were timely paid by the applicant, this International Search Report covers only those claims for which fees were paid, specifically claims Nos.:
4. ☒ No required additional search fees were timely paid by the applicant. Consequently, this International Search Report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

1, 2

Remark on Protest

- ☐ The additional search fees were accompanied by the applicant's protest.
- ☐ No protest accompanied the payment of additional search fees.

International Application No. PCT/ US 03/34558

## FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 218

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

## 1. claims: 1,2

A two integrated-circuit pipeline accelerator.

## 1.1. claim: 3

A reconfigurable pipeline accelerator

## 1.2. claims: 4-14,41-50

Method and apparatus for pipelining data communications and processing in a processor/coprocessor system.

---

## 2. claims: 15,31-34,59

Method and apparatus for configuring a hardwired pipeline.

---

## 3. claims: 16,35-40,60

Exception handling in a processor/co-processor system

---

## 4. claims: 17-23,51-54

Method and apparatus for routing data in a processor/co-processor system.

---

## 5. claim: 24

A multithreaded coprocessor.

---

## 6. claims: 25-30,55-58

Control of the request handling order in a co-processor.

---

## 7. claims: 61-65

A method for designing hardwired pipelines.

---

## INTERNATIONAL SEARCH REPORT

on patent family members

PCT/US 03/34558

Patent document cited in search report		Publication date	Patent family member(s)		Publication date
EP 1061439	A	20-12-2000	EP	1061439 A1	20-12-2000
			EP	1104562 A1	06-06-2001
			WO	0077627 A1	21-12-2000
			JP	2003502728 T	21-01-2003
			US	6782445 B1	24-08-2004
US 5892962	A	06-04-1999	CA	2215598 A1	12-05-1996

Form PCT/IS/40 (to patent family members) (January 2004)

## フロントページの続き

(31)優先権主張番号 10/684,053  
 (32)優先日 平成15年10月9日(2003. 10. 9)  
 (33)優先権主張国 米国(US)  
 (31)優先権主張番号 10/684,057  
 (32)優先日 平成15年10月9日(2003. 10. 9)  
 (33)優先権主張国 米国(US)  
 (31)優先権主張番号 10/684,102  
 (32)優先日 平成15年10月9日(2003. 10. 9)  
 (33)優先権主張国 米国(US)

(81)指定国 AP(BW, GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), EA(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), EP(AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG), AE, AG, AI, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, GR, GU, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW

(74)代理人 100135585

弁理士 西尾 務

(72)発明者 ラーブ、ジョン、ダブリュー、

アメリカ合衆国 バージニア州 20110 マナサッス、リバー クレスト ロード 9350

(72)発明者 ジャクソン、ラリー

アメリカ合衆国 バージニア州 20112 マナサッス、クレストブロック ドライブ 13093

(72)発明者 ジョウンス、マーク

アメリカ合衆国 バージニア州 20120 セントレビル、オークマー プレイス 15342

(72)発明者 カーサロ、トロイ

アメリカ合衆国 バージニア州 22701 カルペパー、ケストラル コート 1524